Universidad Carlos III de Madrid
Escuela Politécnica Superior
Departamento de Tecnología Electrónica

PROYECTO FIN DE CARRERA

INGENIERÍA TÉCNICA DE TELECOMUNICACIONES

TELEMÁTICA

ANÁLISIS DE SEGURIDAD EN COMUNICACIONES USB

AUTOR: RAMÓN BLANCO GONZALO

Director de proyecto: Raúl Sánchez Reíllo

Noviembre 2009

Agradecimientos

Me gustaría dar las gracias a Raúl por darme la oportunidad de trabajar estos meses en un ambiente tan cordial y humano, y por su habilidad de sacar tiempo de donde no lo hay. También me gustaría agradecer a toda la gente del GUTI su apoyo incondicional y el trato inigualable. Chapó.

Por supuesto, agradecer a mi familia y amigos su apoyo constante y el hecho de estar siempre ahí. Mención especial a una persona que soporta todos mis problemas y soporta lo insoportable. Gracias.

Gracias a Fran, Luís, José y José Luís por echarme una mano cuando la pedí y cuando no la pedí.

Gracias a Matthew Dunn, por darme ese empujón que me hacía falta.

Por último, no quiero terminar mi carrera sin agradecer a Eduardo Joyanes, Michael García Lorenz y a Jerónimo Arenas el haber hecho que haya aprendido a levantarme una y otra vez.

Índice de contenidos

Listado de acrónimos

AES: Advanced Encryption Standard.

ACK: Acknowledgement.

CATC: Computer Access Technology Corporation. Es un protocolo que utiliza el analizador de USB que emplearemos en el proyecto.

CRC: Comprobación de Redundancia Cíclica.

DH: Diffie-Hellman (es el protocolo que permite el intercambio secreto de claves entre dos partes que no han tenido contacto previo).

FAR: False Acceptance Rate.

FIFO: First In First Out.

FRR: False Rejection Rate.

FS: Full Speed.

HS: High Speed.

IETF: Internet Engineering Task Force.

JFIF: JPEG File Interchange Format.

JPEG: Joint Photographic Experts Group.

LIFO: Last In First Out.

LS: Low Speed.

MITM: Man-In-The-Middle.

NACK: Negative Acknowledgement.

PID: Packet IDentification.

RSA: Algoritmo criptográfico de cifrado asimétrico creado por Ron Rivest, Adi Shamir y Len Adleman.

SOF: Start Of Frame.

USB: Universal Serial Bus.

Listado de términos en inglés

Hemos realizado este listado con objeto de aclarar ciertos términos que utilizaremos en inglés, ya que consideramos que en este idioma serán más precisos.

Driver: "Un driver o controlador de dispositivo, es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz -posiblemente estandarizada- para usarlo."

Endpoint: "Cada interfaz del conjunto de interfaces (pipe) en las que se organiza un dispositivo USB cuando se conecta a un sistema (y es reconocido y configurado), que especifica que partes del hardware del dispositivo interactúan con el sistema USB"

KeepAlive: "En telecomunicaciones se trata de mensajes enviados entre emisor y receptor con objeto de dar a conocer que se está activo en la comunicación. La traducción literal del inglés sería *mantenerse vivo.*"

Pipe: En USB, "vías de comunicación entre las aplicaciones que se ejecutan en el host (clientes) y los distintos endpoints en los dispositivos USB (servidores)"

PnP: (Plug-and-Play) es la tecnología que permite a un dispositivo informático ser conectado a un ordenador sin tener que configurar (mediante jumpers o software específico proporcionado por el fabricante) ni proporcionar parámetros a sus controladores. Para que sea posible, el sistema operativo con el que funciona el ordenador debe tener soporte para dicho dispositivo.

Pull-Up: "En electrónica se denomina pull-up bien a la acción de elevar la tensión de salida de un circuito lógico, bien a la tensión que, por lo general mediante un divisor de tensión, se pone a la entrada de un amplificador con el fin de desplazar su punto de trabajo."

Timeout: “Mensaje de error que se genera cuando se agota un tiempo de espera”.

Trigger: (o disparador) “Es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación”. En nuestro proyecto utilizaremos triggering cuando capturemos tramas con el analizador y se cumpla una condición por la que queramos terminar el proceso de captura.

.utg: “Extensión de los archivos que han sido generados por el analizador CATC, los cuales contienen código que permite generar tráfico desde el analizador”

Listado de imágenes

Capítulo 1:
Introducción y objetivos

En este capítulo comentaremos las motivaciones de nuestro proyecto, los objetivos que nos hemos planteado y daremos una idea de cómo los vamos a llevar a cabo.

1.1. Introducción

Actualmente los sistemas biométricos están a la cabeza de la tecnología. La biometría es y será en el futuro una pieza importante dentro de nuestra sociedad. Protección y seguridad son palabras clave en la era digital y los sistemas biométricos se decantan como los más firmes candidatos para llevarlas a cabo. Es por tanto necesidad imperiosa contar con una tecnología que nos aporte confianza y que mejore con ello nuestra calidad de vida.

La biometría puede dejar obsoletos sistemas de seguridad que requieran el transporte de objetos (tarjetas, llaves, etc.) o que requieran recordar una clave, ya que la memoria es frágil y los objetos son fácilmente sustraibles o se pueden perder. En cambio, las características biológicas son fiables, seguras y están exentas de cualquier tipo de substracción.

Siendo fieles a la realidad, debemos añadir que los sistemas biométricos avanzados están emergiendo desde hace relativamente poco tiempo y esto da lugar a que no lleguen a una perfección pretendida. Este es un problema que conlleva cierta gravedad dado que un asunto tan serio como es la seguridad no puede dar lugar a grietas. Debe ser a prueba de fallos. Es aquí donde surge la raíz del objeto de estudio de este proyecto, ya que solo los sistemas biométricos que superen todos los ataques y todas las pruebas de calidad podrán ser dignos de ser llamados “seguros”.

1.2. Objetivos del proyecto

Nuestro principal objetivo será, a través de un analizador de tráfico, examinar ciertos productos que se encuentran en el mercado con el objetivo de encontrar sus fallos de seguridad. En la actualidad, la biometría tiende a constituirse como uno de los principales mecanismos de salvaguarda de nuestra seguridad. Siendo un asunto tan trascendente para la sociedad, la biometría debe proporcionar unos grados de seguridad muy altos, de forma que nuestros bienes más preciados no queden desprotegidos.

En nuestro proyecto, acorde con la iniciativa principal del GUTI (Grupo Universitario de Tecnologías de Identificación, perteneciente al departamento de Tecnología Electrónica), intentaremos aportar más información, más datos y más medios a la causa de forma que, sólo a través de buscar los fallos y no los aciertos, algún día lleguen a nuestras manos sistemas biométricos totalmente seguros y fiables. Con este análisis, evaluaremos ciertos productos hasta decidir sin deben ser aptos o no para el uso diario, con el fin de mejorarlos y que sean más seguros.

Según el objeto de estudio, evaluaremos ciertos dispositivos biométricos disponibles en el laboratorio y lo haremos con un analizador de tráfico USB. Veremos a continuación qué tipo de dispositivos utilizaremos, cómo los analizaremos y los resultados obtenidos.

1.3. Contenido del proyecto

En primer lugar y como punto de partida, introduciremos el proyecto y daremos una idea global de cuáles serán los asuntos principales a tratar y los objetivos que perseguimos. Posteriormente, debido a los dispositivos biométricos que utilizaremos, haremos una breve introducción a la biometría, repasaremos sus inicios y a su vez, comentaremos su situación en la actualidad. Luego describiremos cual es el problema que se nos presenta y propondremos una solución que será la que nos sirva de patrón durante el resto del proyecto.

Comentaremos los ataques informáticos más conocidos e indagaremos en las razones que nos llevarán a elegir los ataques de tipo MITM (Man In The Middle) para nuestro proyecto. Dado que el protocolo USB será el que utilizaremos, realizaremos una descripción exhaustiva de éste, desde sus tipos de cableado hasta su funcionamiento global. A su vez, seguiremos los mismos pasos con el analizador y con los dispositivos utilizados (tanto a nivel hardware como a nivel software), tratando más en profundidad las partes que hemos utilizado en el proyecto. También describiremos los ataques que hemos llevado a cabo y por qué, dando de este modo, una idea de los resultados finales del proyecto, las conclusiones sacadas y las posibles mejoras que se pueden realizar.

Capítulo 2:

Estado del Arte

En este apartado hablaremos de la biometría. Comenzaremos con una breve introducción y seguiremos con repaso histórico hasta llegar a la situación actual.

2.1. Introducción a la biometría

Según la Real Academia Española, definimos biometría como:

“*Estudio mensurativo o estadístico de los fenómenos o procesos biológicos.*”

La biometría es el estudio de métodos automáticos para el reconocimiento único de humanos basados en uno o más rasgos conductuales o físicos intrínsecos. El término se deriva de las palabras griegas "bios" de vida y "metron" de medida. La "biometría informática" es la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos o de conducta de un individuo, para “verificar” identidades o para “identificar” individuos. En las tecnologías de la información (TI), la autenticación biométrica se refiere a las tecnologías para medir y analizar las características físicas y del comportamiento humanas.

Las huellas dactilares, las retinas, el iris, los patrones faciales, de venas de la mano o la geometría de la palma de la mano, representan ejemplos de características físicas (estáticas), mientras que entre los ejemplos de características del comportamiento se incluye la firma, el paso y el tecleo (dinámicas). La voz se considera una mezcla de características físicas y del comportamiento, pero todos los rasgos biométricos comparten aspectos físicos y del comportamiento. Un indicador biométrico es alguna característica con la cual se puede realizar biometría. Cualquiera sea el indicador, debe cumplir los siguientes requisitos:

1. *Universalidad:* cualquier persona posee esa característica.
2. *Unicidad:* la existencia de dos personas con una característica idéntica tiene una probabilidad muy pequeña.
3. *Permanencia:* la característica no cambia en el tiempo.
4. *Cuantificación:* la característica puede ser medida en forma cuantitativa.

Estos requisitos sirven como criterio para descartar o aprobar alguna característica como *indicador biométrico*. Luego de seleccionar algún indicador que los satisfaga, es necesario imponer restricciones prácticas sobre el sistema que tendrá como misión recibir y procesar a estos indicadores. Las características básicas que un sistema biométrico debe cumplir pueden expresarse mediante las restricciones que deben ser satisfechas. Ellas apuntan, básicamente, a la obtención de un sistema biométrico con utilidad práctica. Las restricciones antes señaladas apuntan a que el sistema considere:

El rendimiento, que se refiere a la exactitud, la rapidez y la robustez alcanzada en la identificación, además de los recursos invertidos y el efecto de factores ambientales y/u operacionales. El objetivo de esta restricción es comprobar si el sistema posee una exactitud y rapidez aceptable con una necesidad de recursos razonable.

La aceptabilidad, que indica el grado en que la gente está dispuesta a aceptar un sistema biométrico en su vida diaria. Es claro que el sistema no debe representar peligro alguno para los usuarios y debe inspirar "confianza" a los mismos. Factores psicológicos pueden afectar esta última característica. Por ejemplo, el reconocimiento de una retina, que requiere un contacto cercano de la persona con el dispositivo de reconocimiento, puede desconcertar a ciertos individuos debido al hecho de tener su ojo sin protección frente a un "aparato". Sin embargo, las características anteriores están subordinadas a la aplicación específica. De hecho, para algunas aplicaciones el efecto psicológico de utilizar un sistema basado en el reconocimiento de características oculares será positivo, debido a que este método es eficaz implicando mayor seguridad.

La fiabilidad, que refleja cuán difícil es burlar al sistema. El sistema biométrico debe reconocer características de una persona viva, pues es posible crear dedos de látex, grabaciones digitales de voz prótesis de ojos, etc. Algunos sistemas incorporan métodos para determinar si la característica bajo estudio corresponde o no a la de una persona viva. Los métodos empleados son ingeniosos y usualmente más simples de lo que uno podría imaginar.

Por ejemplo, un sistema basado en el reconocimiento del iris revisa patrones característicos en las manchas de éste, un sistema infrarrojo para chequear las venas de la mano detecta flujos de sangre caliente y lectores de ultrasonido para huellas dactilares revisan estructuras subcutáneas de los dedos.

2.2. Historia de la biometría

La biometría no se puso en práctica en las culturas occidentales hasta finales del siglo XIX, pero era utilizada en China desde al menos el siglo XIV. Un explorador y escritor que respondía al nombre de Joao de Barros escribió que los comerciantes chinos estampaban las impresiones y las huellas de la palma de las manos de los niños en papel con tinta. Los comerciantes hacían esto como método para distinguir entre los niños jóvenes.

En Occidente, la identificación confiaba simplemente en la "memoria fotográfica" hasta que Alphonse Bertillon, jefe del departamento fotográfico de la Policía de París, desarrolló el sistema antropométrico que vemos en la **Ilustración 1** (también conocido más tarde como Bertillonage) en 1883. Éste era el primer sistema preciso, ampliamente utilizado científicamente para identificar a criminales y convirtió a la biometría en un campo de estudio. Funcionaba midiendo de forma precisa ciertas longitudes y anchuras de la cabeza y del cuerpo, así como registrando marcas individuales como tatuajes y cicatrices. El sistema de Bertillon fue adoptado extensamente en occidente hasta que aparecieron defectos en el sistema, principalmente problemas con métodos distintos de medidas y cambios de medida. Después de esto, las fuerzas policiales occidentales comenzaron a usar la huella dactilar, esencialmente el mismo sistema visto en China cientos de años antes.

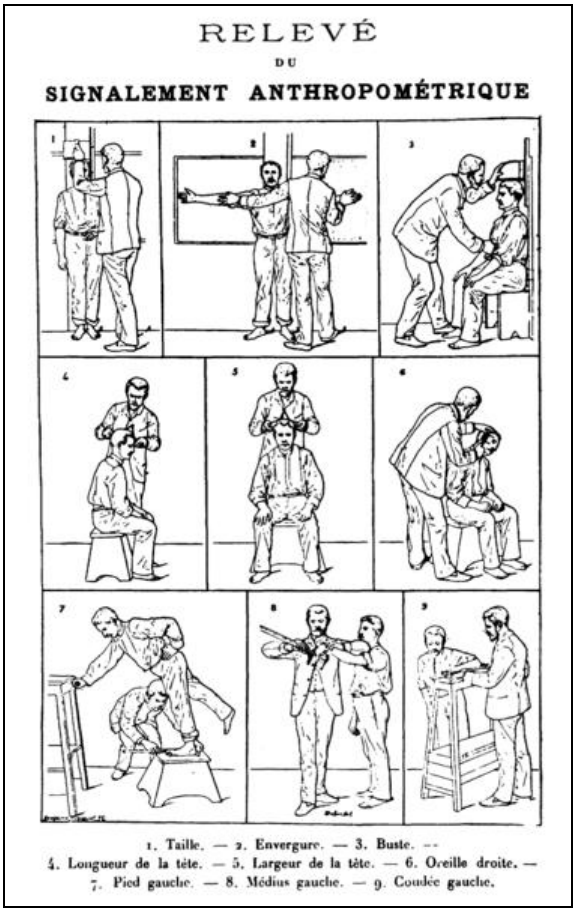


Ilustración 1. Sistema de Bertillon [Ber09]

Un contemporáneo de Bertillon, Sir Francis Galton, definió algunos de los puntos o características desde las cuales las huellas dactilares podían ser identificadas. Estos “puntos Galton”, que vemos en la **Ilustración 2** son la base para la ciencia de identificación por huella dactilar, la cual se ha expandido y efectuado una transición en el pasado siglo.

La identificación por huella digital comienza su transición a la automatización a finales de los años 60 junto con la aparición de las tecnologías de computación. Con la llegada de las computadoras, el subconjunto de los puntos Galton ha sido utilizado para desarrollar la tecnología de reconocimiento automatizado de huellas dactilares.

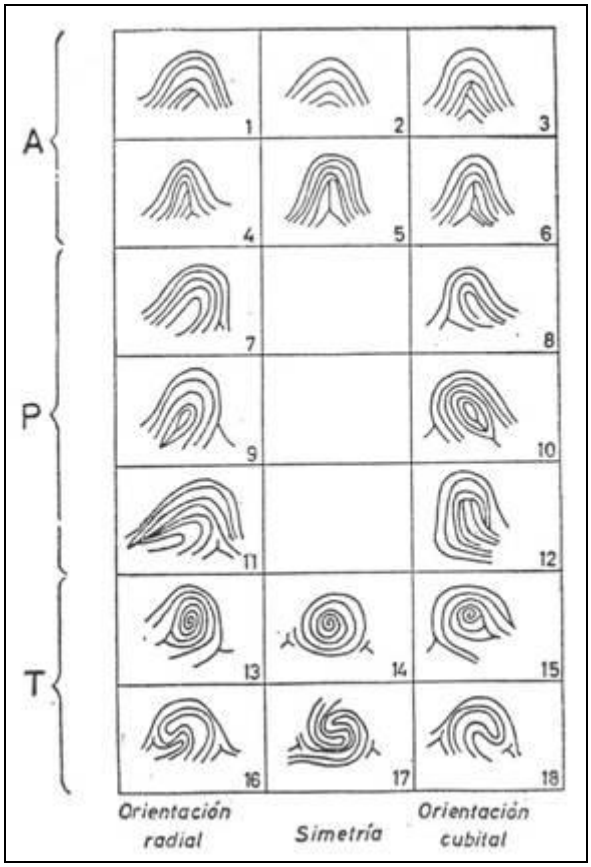


Ilustración 2. Clasificación de Galton [Gal83]

En 1969, hubo un empuje mayor por parte del FBI (Federal Bureau of Investigation) para desarrollar un sistema para automatizar sus procesos de identificación por huellas dactilares, el cual rápidamente se había vuelto abrumador y requería de muchas horas para el proceso manual. El FBI contrató al NBS (National Bureau of Standards), ahora NIST (National Institute of Standards and Technology), para estudiar el proceso de automatización de la clasificación, búsqueda y concordancia de la huellas dactilares. El NIST identifico dos cambios clave: 1 escanear las huellas dactilares y extraer las minucias de cada una y 2 buscar, comparar y combinar las listas de minucias contra grandes repositorios de huellas dactilares.

Si nos referimos a la identificación por iris, la idea para usar patrones de iris como método de identificación fue propuesta en 1936 por el oftalmólogo Frank Burch. Para los 1980's la idea ya había aparecido en películas de James Bond, pero permanecía siendo ciencia ficción.

En 1985 los Doctores Leonard Flom y Aran Safir retomaron la idea. Su investigación y documentación les concedió una patente en 1987. En 1989 Flom y Safir recurrieron a John Daugman para crear algoritmos para el reconocimiento de iris. Estos algoritmos, publicados por Daugman en 1994 y que son propiedad de Iridian Technologies, son la base de todos los productos de reconocimiento de iris.

Hoy en día, la biometría ha crecido desde usar simplemente la huella dactilar a emplear muchos métodos distintos teniendo en cuenta varias medidas físicas (como por ejemplo la huella dactilar o el iris) y de comportamiento (como por ejemplo la voz). Las aplicaciones de la biometría también han aumentado desde sólo identificación hasta importantes sistemas de seguridad.

2.3. Clasificación y funcionamiento de los sistemas biométricos

Como ya hemos comentado anteriormente, excepto la voz que podría considerarse un híbrido, podemos diferenciar los tipos de sistemas biométricos en 2 grupos:

1. Biometría estática: mide la anatomía del usuario. Estos son algunos ejemplos:

    -Huellas dactilares

    -Geometría de la mano

    -Análisis del iris

    -Análisis de retina

    -Venas de la mano

    -Reconocimiento facial

    -Forma de las orejas

2. Biometría dinámica: mide el comportamiento del usuario. Estos son algunos ejemplos:

-Firma manuscrita

-Dinámica de tecleo

-Análisis gestual

Si realizamos una comparativa de los sistemas biométricos más populares, obtenemos una tabla como la siguiente:

	Ojo (Iris)	**Ojo (Retina)**	**Huellas dactilares**	**Geometría de la mano**	**Escritura y firma**	**Voz**	**Cara**
Fiabilidad	Muy alta	Muy alta	Alta	Alta	Media	Baja	Baja
Facilidad de uso	Media	Baja	Alta	Alta	Alta	Alta	Muy alta
Prevención de ataques	Muy alta	Muy alta	Alta	Alta	Media	Media	Baja
Aceptación	Media	Baja	Alta	Muy alta	Muy alta	Alta	Muy alta
Estabilidad	Alta	Alta	Alta	Media	Baja	Media	Media

Ilustración 3. Comparativa sistemas biométricos [Akj04]

En cuanto a la arquitectura de un proceso biométrico, los dispositivos biométricos poseen tres componentes básicos. El primero se encarga de la adquisición analógica o digital de algún indicador biométrico de una persona, como por ejemplo, la adquisición de la imagen de una huella dactilar mediante un escáner. El segundo maneja la compresión, procesamiento, almacenamiento y comparación de los datos adquiridos (en el ejemplo una imagen) con los datos almacenados. El tercer componente establece una interfaz con aplicaciones ubicadas en el mismo u otro sistema. La arquitectura típica de un sistema biométrico se presenta en la **Ilustración 4**. Esta puede entenderse conceptualmente como dos módulos:

1. *Módulo de registro (enrollment module)*
2. *Módulo de identificación (identification module).*

El primero se encarga de adquirir datos relativos al indicador biométrico elegido y entregar una representación en formato digital de éste. El segundo extrae, a partir de la salida del lector, características representativas del indicador. El conjunto de características anterior, que será almacenado en una base de datos central u otro medio como una tarjeta magnética, recibirá el nombre de “patrón”. En otras palabras un patrón es la información representativa del indicador biométrico que se encuentra almacenada y que será utilizada en las labores de identificación al ser comparada con la información proveniente del indicador biométrico en el punto de acceso.

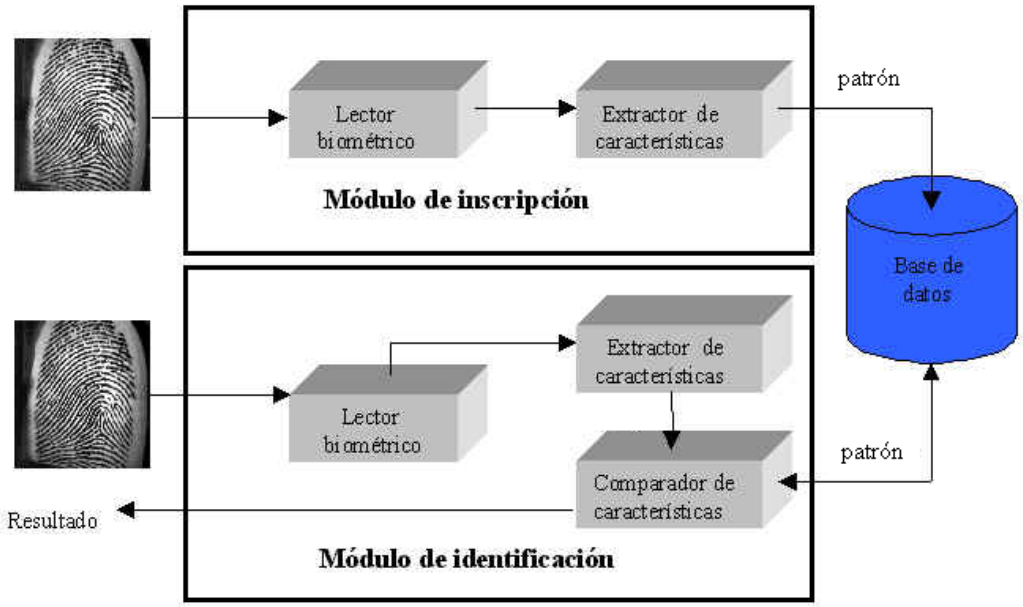


Ilustración 4. Arquitectura del proceso biométrico [Ing00]

El módulo de identificación es el responsable del reconocimiento de individuos, por ejemplo en una aplicación de control de acceso. El proceso de identificación comienza cuando el lector biométrico captura la característica del individuo a ser identificado y la convierte a formato digital, para que a continuación el extractor de características produzca una representación compacta con el mismo formato de los *patrones*. La representación resultante se denomina "vector de características" y es enviada al comparador que confronta a éste con uno o varios *patrones* para establecer la identidad. El conjunto de procesos realizados por el módulo de inscripción recibe el nombre de *fase de registro*, mientras que los procesos realizados por el módulo de identificación reciben la denominación de *fase operacional*. Un sistema biométrico en su fase operacional puede operar en dos modos:

1. *Modo de verificación*

   Un sistema biométrico operando en el modo de verificación comprueba la identidad de algún individuo comparando la característica sólo con los patrones del individuo. Por ejemplo, si una persona ingresa su nombre de usuario entonces no será necesario revisar toda la base de datos buscando el patrón que más se asemeje al de él, sino que bastará con comparar la información de entrada sólo con el patrón que está asociado al usuario. Esto conduce a una comparación uno-a-uno para determinar si la identidad reclamada por el individuo es verdadera o no.

2. *Modo de identificación*

   Un sistema biométrico operando en el modo de identificación descubre a un individuo mediante una búsqueda *exhaustiva* en la base de datos con los patrones. Esto conduce a una comparación del tipo *uno-a-muchos* para establecer la identidad del individuo.

Generalmente es más difícil diseñar un sistema de identificación que uno de verificación. En ambos casos es importante la exactitud de la respuesta. Sin embargo, para un sistema de identificación la rapidez también es un factor crítico. Un sistema de identificación necesita explorar toda la base de datos donde se almacenan los patrones, a diferencia de un sistema verificador.

Una decisión tomada por un sistema biométrico distingue "personal autorizado" o "impostor". Para cada tipo de decisión, existen dos posibles salidas, verdadero o falso. Por lo tanto existe un total de cuatro posibles respuestas del sistema:

1. Una persona autorizada es aceptada
2. Una persona autorizada es rechazada
3. Un impostor es rechazado

4. Un impostor es aceptado.

Las salidas números 1 y 3 son correctas, mientras que las números 2 y 4 no lo son. El grado de confianza asociado a las diferentes decisiones puede ser caracterizado por la distribución estadística del número de personas autorizadas e impostores. En efecto, las estadísticas anteriores se utilizan para establecer dos tasas de errores:

1. Tasa de falsa aceptación, **FAR** (*False Acceptance Rate*), que se define como la probabilidad con que un impostor sea aceptado como un individuo autorizado.
2. Tasa de falso rechazo, **FRR** (*False Rejection Rate*), definida como la probabilidad con que un individuo autorizado sea rechazado como un impostor.

La FAR y la FRR son funciones del grado de seguridad deseado. En efecto, usualmente el resultado del proceso de identificación o verificación será un número real normalizado en el intervalo [0, 1], que indicará el "grado de similitud" o correlación entre la muestra biométrica proporcionada por el usuario y la(s) almacenada(s) en la base de datos. Si, por ejemplo, para el ingreso a un recinto se exige un valor alto para el grado de parentesco (un valor cercano a 1), entonces pocos impostores serán aceptados como personal autorizado y muchas personas autorizadas serán rechazadas. Por otro lado, si el grado de parentesco requerido para permitir el acceso al recinto es pequeño, una fracción pequeña del personal autorizado será rechazada, mientras que un número mayor de impostores será aceptado.

El ejemplo anterior muestra que la FAR y la FRR están íntimamente relacionadas, de hecho son duales una de la otra: una FRR pequeña usualmente entrega una FAR alta, y viceversa. El grado de seguridad deseado se define mediante el umbral de aceptación u, un número real perteneciente al intervalo [0,1] que indica el mínimo grado de parentesco permitido para autorizar el acceso del individuo.

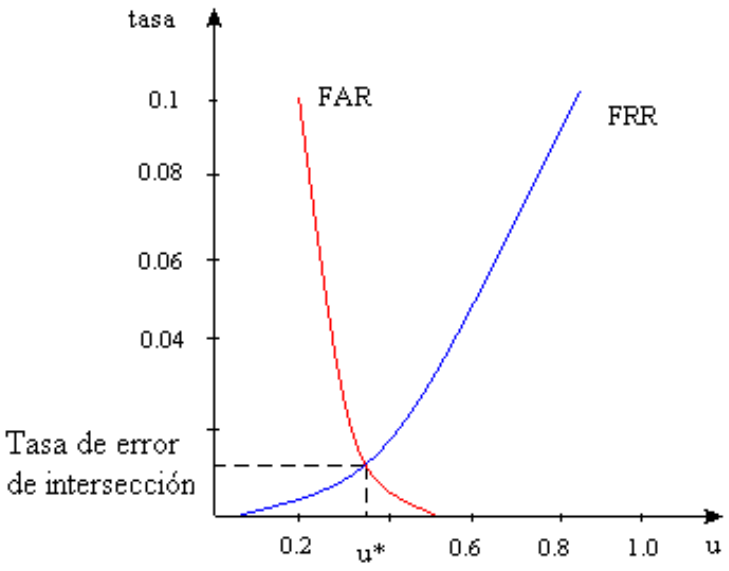


Ilustración 5. Funciones FAR y FRR [Ing00]

La FRR es una función estrictamente creciente y la FAR una estrictamente decreciente en "u". La FAR y la FRR al ser modeladas como función del umbral de aceptación tienen por dominio al intervalo real [0,1], que es además su recorrido, puesto que representan frecuencias relativas. En la **Ilustración 5** se muestra una gráfica típica de la FRR y la FAR como funciones de u. En esta figura puede apreciarse un umbral de aceptación particular, denotado por u*, donde la FRR y la FAR toman el mismo valor. Este valor recibe el nombre de tasa de *error de intersección (cross-over error rate)* y puede ser utilizado como medida única para caracterizar el *grado de seguridad* de un sistema biométrico.

En la práctica, sin embargo, es usual expresar los requisitos de rendimiento del sistema, tanto para verificación como para identificación, mediante la FAR. Usualmente se elige un umbral de aceptación por debajo de u* con el objeto de reducir la FAR, en detrimento del aumento de la FRR.

2.4. Biometría en la actualidad

Hoy en día, la biometría está presente en multitud de situaciones cotidianas y sus aplicaciones abarcan un gran número de sectores: desde el acceso seguro a ordenadores y redes, protección de ficheros electrónicos, hasta el control horario y control de acceso físico a una sala de acceso restringido

Este tipo de reconocimiento se ha convertido en una herramienta habitual en las fuerzas de la policía durante los procesos de investigación criminal, posibilitando la detención de delincuentes a nivel mundial. Se usan para seguridad física en general, controles de inmigración, seguridad de equipaje y carga, vigilancia policial y una cantidad de aplicaciones para prevención del fraude: Acceso a cajeros automáticos, asistencia sanitaria, votación, licencia de conducir, etc.

A pesar de la existencia de sistemas de seguridad biométricos desde hace años, su coste y grado de intrusión los hacía poco atractivos frente a otros sistemas como claves personales o tarjetas. Ahora su coste importa menos, si su precio puede llegar a evitar intrusión en las redes o a salvar vidas humanas en último extremo. Los expertos consideran a la biometría como el sistema de autenticación más fiable, ya que no puede ser transferido, robado u olvidado, y es casi imposible de copiar. Son rasgos inherentes a cada individuo que le hacen único respecto a los demás. A continuación haremos un pequeño esbozo de los principales sistemas biométricos en la actualidad:

La huella dactilar es la tecnología biométrica más madura. Hoy en día, algunos sistemas se basan en el mismo método que usa la policía a través de la minucia, es decir, los puntos distintivos en la forma de los arcos de la huella. Sin embargo, existen otros más sofisticados equipados con biosensores capaces de detectar el calor o el flujo sanguíneo desprendido por los dedos.

En cuanto a la geometría de la mano podemos decir que equilibra fiabilidad con la facilidad de uso en amplios grupos de usuarios. Se están usando en su mayoría para el control de acceso y asistencia al trabajo, tanto en administraciones públicas como en empresas. En España, compañías como Acens **[Ace09]** cuentan con un sistema de este tipo para acceder a su sede. La identificación por escaneo de retina se basa en el análisis de la capa de vasos sanguíneos situada en el globo ocular. El sistema digitaliza el patrón de la retina, único y estable en el tiempo. Hoy en día, se utiliza para el acceso tanto físico como virtual a instalaciones y redes informáticas de alta seguridad.

En el reconocimiento del iris ocular, los más de 200 puntos distintivos de un iris le convierten en uno de los métodos más fiables. Aeropuertos europeos como Heathrow, en Londres, y Schiphol, en Ámsterdam, disponen de un sistema de reconocimiento de iris para la identificación de pasajeros habituales, y en Estados Unidos ya son más de 50 aeropuertos los que usan esta tecnología. **[Ama07]**.

En cuanto al reconocimiento por rostro, se registran las características de la cara mediante una cámara digital que genera una imagen del usuario**.** Ha recibido una buena cantidad de atención en estos últimos años. La gente identifica fácilmente a otras personas por su cara, pero automatizar esta tarea no es para nada sencillo. Mucho del trabajo en esta área se ha dedicado a capturar la imagen facial. Algunas compañías están desarrollando sistemas que identifican individuos por la huella de toda la palma de la mano.

Su uso se ha extendido como sistema de vigilancia en grandes espacios públicos, incluidas las calles de ciudades como Londres o Miami **[Iwr02].** Una de estas plataformas de software es Nemesis, desarrollada por la compañía estadounidense Biovisec, que permite registrar imágenes en 3D de cientos de rostros en espacios públicos.

El reconocimiento por voz esta basado en una autenticación de la señal acústica de voz a través de una compleja tecnología. Su ventaja es que no necesita nuevos dispositivos de hardware y tiene gran aceptación. En cuanto al método de identificación de firma consiste en la verificación de la firma digitalizada. Su principal virtud reside en la aceptación por parte de los usuarios de la firma como un método natural de verificación de la identidad, aunque tiene una fiabilidad menor que otros sistemas.

En España, el uso de los sistemas biométricos no está demasiado extendido. Sólo en algunas áreas de alta seguridad como el acceso a salas de servidores u otras instalaciones sensibles suelen tener un control de acceso biométrico. La tecnología más utilizada es el reconocimiento de huella, por su buena relación coste-implementación y la no intrusión en la operativa. A la extensión del uso de la biometría también está contribuyendo la disminución de los precios del hardware necesario para algunas aplicaciones. Fabricantes de hardware, como HP Compaq o Dell, ofrecen dispositivos biométricos en sus estaciones de trabajo.

Con vistas al futuro, lo único que se puede decir con certeza es que la industria de la biometría está creciendo. Del incremento en las ventas resultará una reducción en los costes, tal y como ha sucedido con la reducción del precio en los ordenadores. Las ventas no son la única parte de la industria biométrica que está creciendo. El número de tecnologías y fabricantes también se está expandiendo. Algunas casas están explorando tecnologías con nuevos atributos fisiológicos para identificación, mientras que otras están mejorando tecnologías actualmente en uso **[PC08].**

Capítulo 3:

Problema y solución

En este apartado haremos una descripción del problema, proponiendo una solución al mismo. Comentaremos el proceso de análisis que realizaremos y los dispositivos USB que hemos utilizado.

3.1. Descripción del problema

Nos enfrentamos a un proceso de análisis de calidad de distintos dispositivos biométricos, luego tendremos que decidir en primer lugar cuáles van a ser esos dispositivos, cómo los vamos a analizar, qué tipo de ataques vamos a llevar a cabo y cómo lo integraremos todo.

Los dispositivos que utilizaremos van a venir limitados por el analizador que tenemos disponible, ya que en este caso, nos interesa realmente examinar la capacidad y las prestaciones que va a desarrollar el nuevo equipo adquirido en el laboratorio. En esta tesitura, los dispositivos a utilizar serán del tipo USB.

De entre la multitud de dispositivos biométricos que se nos presentan con estas características, debemos seleccionar cuales serán los más adecuados según nuestros intereses. Ya que en el laboratorio disponemos de una gran variedad de ellos, no tendremos limitaciones de ese tipo. Por otra parte, nos interesa examinar aquellos dispositivos que sean más populares y que tengan mayor aceptación global, es decir, los más universales.

El modo de análisis se ceñirá a los dominios del laboratorio. Como ya hemos comentado, nos interesa probar el nuevo equipo de analizadores de tráfico adquirido por lo que nuestra solución ha de estar forzosamente basada en el nuevo material. El análisis de los dispositivos debe ser conciso y debe ajustarse a ciertos parámetros que sean reveladores de la calidad de los dispositivos elegidos. De entre todas las formas de examinar estos parámetros deberemos escoger aquellas que estén a nuestro alcance por coste, duración, simplicidad y universalidad.

Según los criterios de los análisis biométricos y sabiendo que los factores ambientales no van a ser algo que debamos tener en cuenta (las condiciones físicas en el laboratorio son las idóneas), ya tenemos el problema acotado a falta de integrar el conjunto en la solución final.

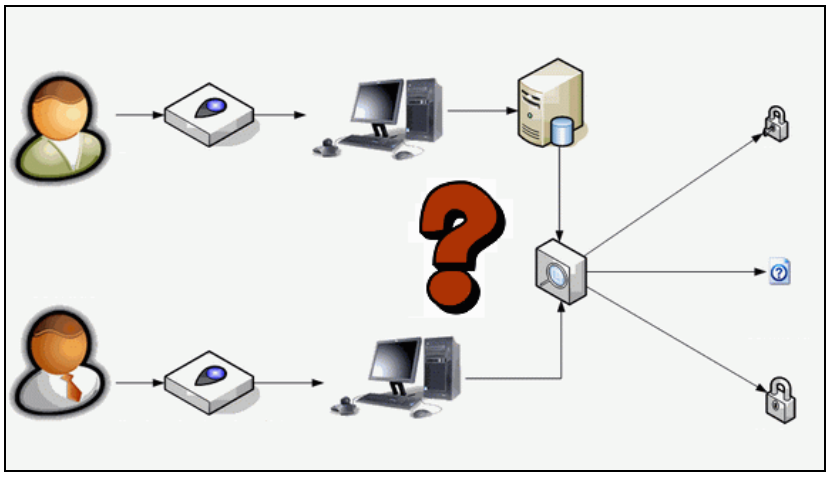


Ilustración 6. Modo de análisis de un dispositivo biométrico

3.2. Solución propuesta

Según las cotas que nos hemos marcado al principio, la solución propuesta y llevada a cabo tiene las siguientes características:

3.2.1. Proceso de análisis

El proceso de análisis será llevado a cabo con el analizador CATC de la empresa Lecroy. Hemos escogido este analizador por varios motivos, entre los que destaca la capacidad de analizar y generar tramas de tipo USB.

Entre otras características, también soporta diversas velocidades (USB 1.1, USB 2.0), permite realizar triggering de diferentes formas y permite capturar eventos en tiempo real.

Ilustración 7. Analizador CATC

El analizador en cuestión tiene todas las características que hemos requerido en un principio y su versatilidad nos va a permitir utilizarlo en un futuro para multitud de tareas. Por otra parte, el hecho de que este especializado en comunicaciones USB lo hace aún más acorde a nuestras necesidades, ya que la comunicación USB está muy extendida en la actualidad.

El proceso de análisis se realizará sobre las comunicaciones “dispositivo USB - Host”, siendo de gran trascendencia un manejo óptimo del software de todo el hardware implicado. Por simplicidad y a la vista de que los resultados obtenidos van a ser los mismos, hemos decidido utilizar el Host que soporta el software del analizador (el que se encargará de la monitorización de las comunicaciones), también como Host implicado en el proceso de comunicación con los dispositivos USB. De esta forma, la complejidad de la arquitectura se reduce considerablemente, lo que nos facilitará un mayor radio de acción en el trabajo y a la postre una mayor comodidad.

3.2.2. Dispositivos utilizados

Teniendo en cuenta las características del analizador elegido, los dispositivos utilizados serán de tipo USB. Siguiendo los dictámenes propuestos, éstos tendrán que estar disponibles en el laboratorio, ser ampliamente conocidos y utilizados, y además su uso debe ser sencillo y compatible con el analizador (tanto por razones de hardware como de software). Es por estas razones por las que nos hemos decantado por utilizar dispositivos de reconocimiento de iris, de huella dactilar y de venas de la mano.

Para el **reconocimiento de iris** hemos utilizado una cámara de Panasonic. Dicha cámara permite grabar tanto imágenes como video. Para nuestro fin utilizaremos esta cámara exclusivamente para tomar imágenes del ojo.

Para el **reconocimiento de huella dactilar** nos hemos basado en 2 dispositivos:

Por una parte, hemos escogido el lector de huellas digitales externo de DELL, que tiene un sistema de lectura de huella dactilar por barrido (deslizando el dedo por una superficie pequeña y rectangular).

El segundo dispositivo elegido es el escáner de huella digital de Biometrika, que permite a su vez a utilización de Smart-Cards, aunque no las usaremos para nuestros propósitos.

Para el reconocimiento de **venas de la mano** hemos utilizado el dispositivo de Fujitsu, PalmSecure, que obtiene una imagen del mapa de venas de la mano a través de un sistema de infrarrojos.

3.2.3. Proceso de análisis de los dispositivos

Dichos dispositivos serán objeto de diferentes análisis:

En primer lugar, nos familiarizaremos con el funcionamiento propio de cada dispositivo, autenticándonos, viendo cómo almacenan la información biométrica y los registros y comprobando cuáles son sus límites. Posteriormente y ya con el analizador de por medio, se examinará la información inicial que transmiten los dispositivos con el Host para tener clara la forma que tienen de comunicarse, los paquetes que envían, etc. Así comprenderemos mejor como se producen las comunicaciones Dispositivo-Host utilizando USB.

El siguiente paso será el de analizar la información biométrica que capturan dichos dispositivos. Ésta será transmitida en forma de datos (junto con estructuras de control, etc.), los cuáles debemos delimitar y extraer con el fin de comprender mejor el funcionamiento del sistema y como parte de los ataques que explicaremos después.

Conociendo las características, el formato de las tramas, el tamaño de los datos, los tiempos de espera y las debilidades de cada dispositivo, será entonces cuando comencemos a realizar ataques de tipo MITM: suplantación de identidad, ataques de sustitución, etc. Explicaremos en el siguiente capítulo el por qué de la realización de estos ataques y no otros.

Tras realizar los ataques pertinentes, haremos un cómputo global de todos ellos, con objeto de dar a conocer sus pros y sus contras, y también de hacer un análisis de las prestaciones del analizador utilizado.

Capítulo 4:

Tipos de Ataques informáticos

En este apartado comentaremos diferentes ataques informáticos que se podrían llevar a cabo en el proyecto. Hablaremos de cada uno de ellos y al final haremos una justificación sobre la decisión de cual utilizaremos. Un ataque informático es un método por el cual un individuo, mediante un sistema informático, intenta tomar el control, desestabilizar o dañar otro sistema informático (ordenador, red privada, etc.).

4.1. Ataque Man In The Middle (MITM)

En criptografía, un ataque MITM (o *intermediario*, en castellano) es un ataque en el que el enemigo adquiere la capacidad de leer, insertar y modificar a voluntad, los mensajes entre dos partes sin que ninguna de ellas conozca que el enlace entre ellos ha sido violado. El atacante debe ser capaz de observar e interceptar mensajes entre las dos víctimas. La posibilidad de un ataque de intermediario sigue siendo un problema potencial de seguridad serio, incluso para muchos criptosistemas basados en clave pública. Existen varios tipos de defensa contra estos ataques MITM que emplean técnicas de autenticación basadas en claves públicas, autenticación mutua fuerte, claves secretas o contraseñas.

La integridad de las claves públicas debe asegurarse de alguna manera, pero éstas no exigen ser secretas, mientras que las contraseñas y las claves de secreto compartido tienen el requisito adicional de la confidencialidad. Las claves públicas pueden ser verificadas por una autoridad de certificación (CA), cuya clave pública sea distribuida a través de un canal seguro (por ejemplo, integrada en el navegador web o en la instalación del sistema operativo).

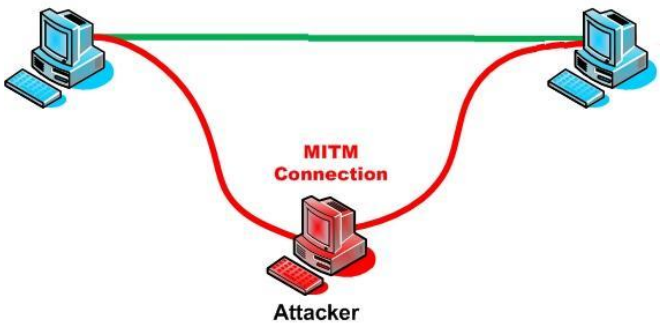


Ilustración 8. Ataque MITM [Ow09]

Ataque de Replay

Un ataque de replay es una forma de ataque de red, en el cual una transmisión de datos válida es maliciosa o fraudulentamente repetida o retardada. Es llevada a cabo por el autor o por un adversario que intercepta la información y la retransmite, posiblemente como parte de un ataque enmascarado. El ataque de replay pretende capturar información y posteriormente reenviarla con el objetivo de falsificar la identidad de uno de los lados. Es un hecho muy conocido que el intercambio de claves DH (*Diffie-Hellman*) de manera primitiva adolece de ser atacado por un ataque de replay, sin embargo, esto puede evitarse con una marca secuencial o una marca de tiempo. Hoy día ninguna aplicación que use el esquema de intercambio de claves DH de manera adecuada se ve afectada por este ataque.

4.2. Ataque de denegación de servicio

En seguridad informática, un ataque de denegación de servicio, también llamado ataque **DoS** (*Denial of Service*), es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima. Se genera mediante la saturación de los puertos con flujo de información, haciendo que el servidor se sobrecargue y no pueda seguir prestando servicios, por eso se le dice "denegación", pues hace que el servidor no dé abasto a la cantidad de usuarios. Esta técnica es usada por los llamados crackers para dejar fuera de servicio a servidores objetivo.

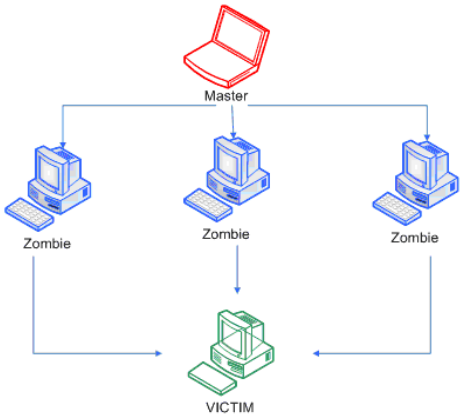


Ilustración 9. Ataque DDos [Ow09]

El llamado **DDoS** (siglas en inglés de *Distributed Denial of Service*, denegación de servicio distribuida) es una ampliación del ataque DoS y se efectúa con la instalación de varios agentes remotos en muchas computadoras que pueden estar localizadas en diferentes puntos. El invasor consigue coordinar esos agentes para así, de forma masiva, amplificar el volumen de saturación de información, pudiendo darse casos de un ataque de cientos o millares de computadoras dirigido a una máquina o red.

Esta técnica se ha revelado como una de las más eficaces y sencillas a la hora de colapsar servidores. La tecnología distribuida ha ido sofisticándose hasta el punto de otorgar poder de causar daños serios a los ordenadores de personas con escaso conocimiento técnico. En ocasiones, esta herramienta ha sido utilizada como un método para comprobar la capacidad de tráfico que un ordenador puede soportar sin volverse inestable y perjudicar los servicios que desempeña. Un administrador de redes puede así conocer la capacidad real de cada máquina.

4.3. Ataque de día cero

Un ataque o amenaza de día-cero (o también hora-cero) es un ataque contra un ordenador, a partir del cual se intentan explotar determinadas vulnerabilidades de algún programa o programas que corren bajo el mismo. Normalmente estas vulnerabilidades son desconocidas, no están publicadas o no existe un parche que las solvente. El término día cero se utiliza también para describir virus desconocidos o virus de día-cero. Los ataques de día-cero se liberan antes de que el vendedor del parche lo libere públicamente. Este tipo de ataques circulan generalmente entre las filas de los potenciales atacantes hasta que finalmente son liberados en foros públicos.

Un ataque de día-cero normalmente es desconocido para la gente y el vendedor del producto. Los ataques día-cero ocurren cuando una vulnerabilidad tiene una ventana de tiempo existente entre el tiempo en el que se publica una amenaza y el tiempo en el que se publican los parches que los solucionan. Normalmente estos parches los preparan los propios responsables del programa "defectuoso" en cuestión.

4.4. Tipo de ataque elegido

El tipo de ataque elegido será el ataque MITM por varias razones. En primer lugar, es el único que cumple todos los requisitos del proyecto, proporcionándonos la información transmitida por los dispositivos y permitiéndonos modificarla y suplantar identidades. También va acorde con el analizador. De hecho el analizador es la herramienta perfecta para llevar a cabo este tipo de ataques (tiene la capacidad de capturar tráfico y de generarlo).

Los demás tipos de ataques vistos, quizá también nos puedan servir para comprobar ciertos rasgos de seguridad de los dispositivos, pero no están en el ámbito de este proyecto, por lo que los desecharemos (los ataques DoS o los ataques de día cero no nos proporcionarán datos relevantes). Los ataques que realizaremos serán varios.

En primer lugar, colocaremos el analizador entremedias del PC (que funciona como Host) y del dispositivo a analizar. Capturaremos toda la información que se intercambien y la recopilaremos para analizarla. En este paso obtendremos imágenes en claro o no, dependiendo de los sistemas de seguridad de los que dispongan los dispositivos.

En segundo lugar, trabajaremos con esa información en dos líneas diferentes. Por un lado comprobaremos si podemos realizar el ataque directamente enviando al destino los datos obtenidos y por otro lado modificaremos esos datos con objeto de cumplir diferentes propósitos que tengamos en el proyecto con el fin de evaluar la seguridad de los dispositivos. De este modo se verán completados los ataques MITM, llegando en última instancia a suplantar identidades en los dispositivos biométricos utilizados.

Capítulo 5:

USB. Formato, protocolo y funcionamiento

En este capítulo haremos un repaso al protocolo USB. Comentaremos sus principales características, tanto a nivel físico como a nivel lógico. Por último resumiremos el proceso de chirp handshake, provocado por una incompatibilidad de velocidades entre ciertos dispositivos en nuestro proyecto.

5.1. Características generales

El sistema de comunicación utilizado para el proyecto ha sido USB. ***Universal Serial Bus*** (bus universal en serie).

Ilustración 10. Logotipo USB [Usb09]

Las razones para utilizar este estándar son muchas y variadas. Van desde la simplicidad de sus conexiones, pasando por los tipos de velocidades que soporta, hasta la popularidad que tiene hoy en día la especificación USB. El bus universal en serie, consiste en una norma para bus periférico, desarrollado tanto por empresas de informática como de telecomunicación. USB permite adjuntar dispositivos periféricos al ordenador rápidamente, sin necesidad de reiniciarlo ni de volver a configurar el sistema. Los dispositivos con USB se configuran automáticamente tan pronto como se han conectado físicamente. Además, se pueden unir dispositivos USB en una cadena para conectar más de dos dispositivos al ordenador mediante otros periféricos USB.

Las siglas USB corresponden a *Universal Serial Bus* (Bus Serie Universal) por lo que como su nombre indica, se trata de un sistema de comunicación entre dispositivos electrónicos − informáticos que sólo transmite una unidad de información a la vez. Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- Baja velocidad LS (*Low Speed*, USB 1.0): Tasa de transferencia de hasta 1,5 Mbps (192 KB/s) o más dependiendo la subversión. Utilizado en su mayor parte por dispositivos de interfaz humana como los teclados, los ratones, hornos microondas y artículos del hogar.

- Velocidad completa FS (*Full Speed*, USB 1.1): Tasa de transferencia de hasta 12 Mbps (1'5 MB/s), según este estándar. Ésta fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de impedancias LIFO (Last In First Out).

- Alta velocidad HS (*High Speed*, USB 2.0): Tasa de transferencia de hasta 480 Mbps (60 MB/s) pero por lo general de hasta 125Mbps (16MB/s).

- Super alta velocidad (USB 3.0): Actualmente se encuentra en fase experimental y tiene una tasa de transferencia de hasta 4.8 Gbps (600 MB/s) **[Usb3.0].**

En cuanto a su estructura interna, el USB organiza el bus en forma de árbol descendente, con múltiples dispositivos conectados a un mismo bus, en la que unos elementos especiales, llamados *hubs*, encaminan las señales desde un dispositivo al Host o viceversa.

Primero está el controlador del bus. Éste es el interfaz entre el bus USB y el bus del ordenador. De él cuelgan los dispositivos USB. Los hubs, como son un dispositivo USB más, también cuentan. A un hub se puede conectar uno o más dispositivos, que a su vez pueden ser otros hubs, así tenemos varios dispositivos conectados a un solo controlador que pueden ser como máximo 127.

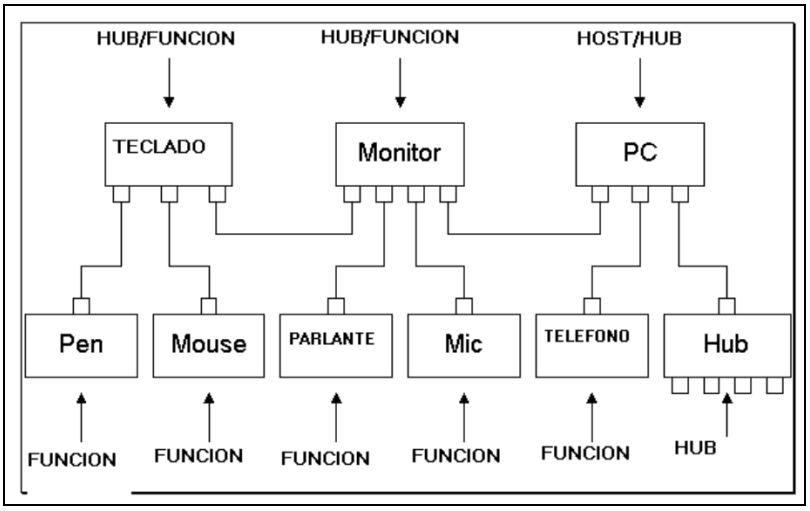


Ilustración 11. Hub USB [Nec09]

Es conveniente resaltar que todos los dispositivos deben seguir reglas de comportamiento básicas, estandarizadas. Por tanto, todos los dispositivos se configuran de la misma forma, y es mucho más fácil gestionar los recursos que proporcionan; sin embargo, esto no significa que todos los dispositivos son iguales, sino, que todos tienen un sistema de configuración idéntico. Además, el hecho de que no tengamos que tocar (inicialmente) nada en el hardware del dispositivo en sí y que todo sea configurable por software nos lleva a la llamada tecnología PnP (Plug-and-Play).

Todo dispositivo USB tiene la capacidad de ser conectado al ordenador en pleno funcionamiento, sin tener que reiniciarlo, además la configuración del dispositivo nuevo es inmediata y completamente transparente al usuario, tras lo cual el dispositivo está listo para ser empleado sin tocar un tornillo, ni jumpers, canales, etc.

Dentro de la terminología USB, el PC que soporta este tipo de bus, se denomina **Host USB**; mientras que por su parte y dentro de la misma terminología, todo periférico y/o dispositivo, se denomina **Dispositivo USB.**

5.1.1. Host USB:

A diferencia de los dispositivos y los hubs, existe tan solo un host dentro del sistema USB, que como ya dijimos es el ordenador mismo, particularmente una porción del mismo denominado Controlador USB del Host. Este tiene la misión de hacer de interfaz entre el ordenador y los diferentes dispositivos. Existen algunas particularidades respecto a este controlador. Su implementación es una combinación de hardware y software. Puede proveer de uno o dos puntos de conexión iniciales, denominados Hub raíz, a partir de los cuales y de forma ramificada irán conectándose los periféricos. El Host es responsable a nivel de hardware, de los siguientes aspectos dentro del sistema USB:

- Detectar tanto la conexión de nuevos dispositivos USB al sistema como el reconocimiento de aquellos ya conectados, y por supuesto, configurarlos y ponerlos a disposición del usuario, tarea que involucra acciones por software.
- Administrar y controlar el flujo de datos entre el Host y los dispositivos USB, es decir el movimiento de información generada por el mismo usuario.
- Administrar y regular el control de flujo entre el Host y los dispositivos USB, es decir la información que se mueve con el objeto de mantener el orden dentro de los elementos del sistema.
- Recolectar y resumir estadísticas de actividad y estado de los elementos del sistema.
- Proveer de una cantidad limitada de energía eléctrica para aquellos dispositivos que pueden abastecerse con tan solo la energía procedente del computador (teclado, ratón son dos ejemplos claros).

Por otra parte, **a nivel de software** las funciones del controlador de Host se incrementan y complican:

- Enumeración y configuración de los dispositivos conectados al sistema.
- Administración y control de transferencias síncronas de información.
- Administración y control de transferencias asíncronas.
- Administración avanzada de suministro eléctrico a los diferentes dispositivos.
- Administración de la información del bus y los dispositivos USB.

5.1.2. Dispositivos USB:

Dentro de la terminología USB, todos los dispositivos que pueden ser conectados a este bus, a excepción de los hubs, se denominan dispositivos USB. Son dispositivos USB típicos: el ratón, el monitor, modem, etc. Estos dispositivos son capaces de recibir y transmitir información, ya sea del usuario o de control. El común denominador de todas las funciones USB es su cable y el conector del mismo, diseñado y fabricado de acuerdo a las especificaciones del bus, por lo que no cabe preocuparse por la compatibilidad entre equipos de diferentes fabricantes.

Ilustración 12. Entrada USB [Usb09]

Se había hablado de algunos beneficios que esta tecnología entregaba tanto al usuario como a las empresas fabricantes, pero las características de USB son muchas más:

- Todos los dispositivos USB deben tener el mismo tipo de cable y el mismo tipo de conector, más allá de la función que cumplan.
- Los detalles de consumo y administración eléctrica del dispositivo deben ser completamente transparentes para el usuario.
- El computador debe identificar automáticamente un dispositivo agregado mientras opera, y por supuesto configurarlo.
- Los dispositivos pueden ser desconectados mientras el computador está en uso.
- Deben poder compartir un mismo bus tanto dispositivos que requieren de unos pocos Kbps como los que requieren varios Mbps.
- Hasta 127 dispositivos diferentes pueden estar conectados simultáneamente y operando con un mismo ordenador sobre el Bus Serial Universal.
- El bus debe permitir periféricos multifunción, es decir aquellos que pueden realizar varias tareas a la vez, como lo son algunas impresoras que adicionalmente son fotocopiadoras y máquinas de fax.
- Capacidad para manejo y recuperación de errores producidos por un dispositivo cualquiera.
- Soporte para la arquitectura PnP.
- Bajo costo.
- No se necesita un cable extra de alimentación − la mayoría de los periféricos USB obtienen la alimentación del bus USB, con lo cual no requieren un cable de alimentación adicional.
- Más rápido: USB transfiere los datos 10 veces más rápido que los puertos serie tradicionales.

Existen dos tipos de conectores dentro del Bus Serial Universal como podemos ver en la **Ilustración 13**.

Ilustración 13. Conectores USB [Usb09]

El conector Serie A está pensado para todos los dispositivos USB que trabajen sobre plataformas de PC's. Serán bastante comunes dentro de los dispositivos listos para ser empleados con Host PC's, y lo más probable es que tengan sus propios cables con su conector serie A. Sin embargo, esto no se dará en todos los casos, existirán dispositivos USB que no posean cable incorporado, para los cuales el conector Serie B será una característica. Sin embargo este no es un problema, ya que ambos conectores son estructuralmente diferentes e insertarlos de forma errónea será imposible por la forma de las ranuras.

La forma física en la que los elementos se interconectan dentro del sistema USB, puede asemejarse a la topología en estrella. El centro de cada estrella es un hub, un dispositivo que por un lado se conecta al computador o a otro hub y por otro lado, permite conectar al mismo varios dispositivos o en su defecto nuevos hubs. Esta disposición significa que los ordenadores con soporte para USB han de tener tan solo uno o dos conectores USB, pero ello no representa poder contar con tan solo dos dispositivos de esta clase. Muchos dispositivos USB han de traer conectores USB adicionales incorporados, por ejemplo un monitor puede tener 3 ó 4 conectores USB donde pueden ir el teclado, el ratón, y algún otro dispositivo. Por su parte el teclado puede tener otros más, y así sucesivamente hasta tener 127 dispositivos, todos funcionando simultáneamente.

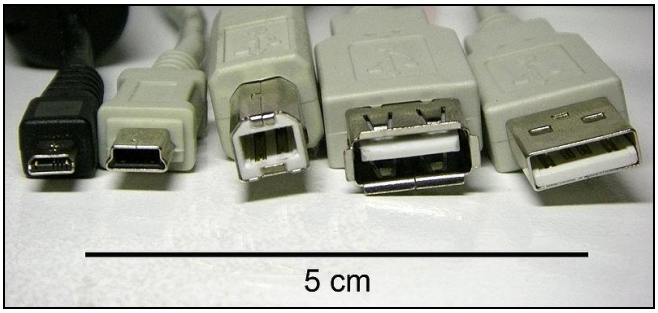


Ilustración 14. Diferentes conectores USB [Usb09]

5.1.3. USB 2.0

La extensión de USB que más utilizaremos será la 2.0, debido a que la mayoría de los dispositivos que utilizamos serán de este tipo. Esta especificación de USB fue lanzada en abril de 3000. También es conocido como USB de alta velocidad y, como hemos comentado antes, soporta tasas de transferencia de datos de hasta 480Mbps. Se trata de una extensión del USB 1.1, utiliza los mismos cables y conectores y es completamente compatible hacia atrás.

Ilustración 15. Logo USB 2.0 [Usb09]

5.2. Protocolo USB

En el protocolo USB toda transferencia de datos o transacción que emplee el bus, involucra al menos tres paquetes de datos. Cada transacción se da cuando el Controlador de Host decide qué dispositivo hará uso del bus, para ello envía un paquete al dispositivo específico. Cada uno de los mismos tiene un número de identificación, otorgado por el Controlador de Host cuando el ordenador arranca o bien cuando un dispositivo nuevo es conectado al sistema. De esta forma, cada uno de los periféricos puede determinar si un paquete de datos es o no para sí. Técnicamente este paquete de datos se denomina Token Packet.

Una vez que el periférico afectado recibe el permiso de transmitir, arranca la comunicación y sus tareas específicas; el mismo informará al Host con otro paquete que ya no tiene más datos que enviar y el proceso continuará con el siguiente dispositivo. Este protocolo tiene un sistema muy eficiente de recuperación de errores, empleando uno de los modelos más conocidos como es el CRC (Comprobación de Redundancia Cíclica). Y puede estar implementado al nivel de software y/o hardware de manera configurable. De hecho si el control es a nivel de hardware, no vale la pena activar el control por software, ya que sería duplicar tareas innecesariamente.

USB divide el tiempo en espacios de 1 ms denominados Tramas, durante las cuales se llevan a cabo las comunicaciones a través de Transacciones que se componen a su vez de Paquetes. Las Transacciones se componen de 3 fases: Token, Dato y Handshake:

- La **fase de Token** se compone de un paquete de Token enviado por el Controlador USB, y siempre está presente en toda transacción. El paquete contiene los campos:

    - PID (identifica el tipo de paquete). Todos los PIDs van protegidos por bits redundantes.

- Dirección del elemento destino (7 bits de dispositivo + 4 bits de elemento interno al dispositivo).
- Control de error CRC de 5 bits (CRC5)

- La **fase de Datos** (opcional) se compone de los paquetes de datos que se transfieren entre el Controlador USB y el dispositivo. Cada paquete se compone de los campos PID, Datos, y CRC16.

Tipo PID	Nombre PID	PID	Descripción
Token	OUT	0001B	Dirección + número de endpoint en una transacción host a función.
	IN	1001B	Dirección + número de endpoint en una transacción función a host.
	SOF	0101B	Indicador de inicio de frame (Start Of Frame) y número de frame.
	SETUP	1101B	Dirección + número de endpoint en una transacción host a función para realizar un Setup de una tubería de control.
Data	DATA0	0011B	PID de paquete de datos par.
	DATA1	1011B	PID de paquete de datos impar.
	DATA2	0111B	PID de paquete de datos de alta velocidad, elevado ancho de banda en una transferencia isócrona en un microframe.
	MDATA	1111B	PID de paquete de datos de alta velocidad para split y elevado ancho de banda en una transferencia isócrona.
Handshake	ACK	0010B	El receptor acepta el paquete de datos libre de errores.
	NAK	1010B	El dispositivo receptor no puede aceptar los datos o el dispositivo emisor no puede enviar los datos.
	STALL	1110B	Endpoint sin servicio o una petición de control sobre una tubería no está soportado.
	NYET	0110B	Aún no se ha recibido una respuesta del receptor.
Special	PRE	1100B	(Token) Habilita trafico de bajada por el bus a dispositivos de velocidad baja.
	ERR	1100B	(Handshake) Error de transferencia split .
	SPLIT	1000B	(Token) Transferencia de alta velocidad split .
	PING	0100B	(Token) Control de flujo sobre endpoints de tipo control o bulk.
	Reservado	0000B	PID reservado.

Ilustración 16. Tipos de fases en las transacciones USB [Usb09]

- La **fase de Handshake** (opcional) se usa para indicar el resultado de la Transacción, así como para indicar si se necesita más tiempo para la contestación. Se compone sólo de un campo PID.

Adicionalmente, el Controlador USB indica el principio de cada Trama y la transmisión hacia dispositivos LS mediante tokens especiales. Además, como vemos en la **Ilustración 16**, en la fase de token se produce el envío de SOF (Start Of Frame), que sirve como indicador de inicio de trama. Veremos que en las diferentes capturas de tráfico USB que realicemos serán muy comunes los paquetes SOF.

5.2.1. Tipos de transferencia de datos

USB soporta 4 tipos de transferencias de datos:

- **Control**, para configuración y control de dispositivos y para manejo del bus.

- **Isócrono**, para transmisión de información con ancho de banda y latencia garantizados, necesario para aplicaciones como audio, telefonía y vídeo. Permite una comunicación periódica y continua entre el sistema y el dispositivo.

- **Interrupción**, para transferencias de pocos datos, no periódicas, de baja frecuencia pero con unos ciertos límites de latencia.

- **Bulk**, para transferencias de grandes cantidades de datos con dispositivos asíncronos, como impresoras, escáneres, cámaras de fotos (foto fija), etc.

5.2.2. Transferencias de Control

Se desarrollan en 3 Transacciones:

- **Transacción de Configuración** (Setup), en la que se envía al dispositivo un paquete que especifica la operación a ejecutar. Ocupa 8 bytes.

- **Transacciones de Datos**, en las que se transfieren los paquetes de datos en el sentido indicado por la Transacción de Configuración. La información útil por paquete puede ser de 8, 16, 32 ó 64 bytes para endpoints FS, y de 8 bytes para endpoints LS.

- **Transacción de Estado**, en la que el receptor informa del estado final de la operación.

Se procesan por medio de un mecanismo "best effort", según el cual el Controlador USB las va procesando en función del tiempo disponible en cada Trama. Como mínimo se reserva el 10% del tiempo de Trama, y se puede utilizar tiempo adicional siempre que las necesidades de los tráficos isócrono y de interrupción lo permitan.

5.2.3. Transferencias Isócronas

- Sólo son utilizables por dispositivos FS.
- La información útil por paquete puede oscilar entre 1 y 1,023 bytes.
- En cada Trama se transfiere un paquete por cada conexión isócrona establecida.

- El sistema puede asignar como máximo el 90% del tiempo de Trama para transferencias isócronas y de interrupción. Si el sistema ya tiene asignado un tiempo de Trama de forma que no garantiza tiempo suficiente como para manejar una nueva conexión isócrona (transmitir un nuevo paquete por Trama), simplemente no se establece la conexión.
- Los posibles errores no se recuperan (la información que no llega a su tiempo, se descarta).

5.2.4. Transferencias de Interrupción

- Aseguran una transacción (paquete) dentro de un periodo máximo (los dispositivos FS pueden solicitar entre 1 y 255 ms, y los LS entre 10 y 255 ms de periodo máximo de servicio).
- Incorpora detección de errores y retransmisión de datos.
- La información útil por paquete puede oscilar entre 1 y 64 bytes para dispositivos FS y entre 1 y 8 bytes para dispositivos LS.
- El sistema puede asignar como máximo el 90% del tiempo de Trama para transferencias isócronas y de interrupción. Si el sistema no puede garantizar tiempo suficiente como para manejar una nueva conexión de interrupción (transmitir un nuevo paquete dentro del periodo máximo requerido), simplemente no se establece la conexión.

5.2.5. Transferencias Bulk

- Sólo son utilizables por dispositivos FS.
- Se procesan por medio de un mecanismo "good effort", en el que el sistema aprovecha cualquier ancho de banda disponible y en el momento en que esté disponible (en otras palabras, no se garantiza una latencia ni un ancho de banda mínimos). Se puede utilizar el tiempo de Trama reservado y no consumido por transferencias de Control (10%).
- Incorporan mecanismos de control de errores para garantizar la entrega de datos.
- La información útil por paquete puede ser de 8, 16, 32 ó 64 bytes.

Estos 4 tipos de transferencias están disponibles como interfaces software que el sistema pone a disposición de los drivers de dispositivo, estando los drivers obligados a comunicarse con los dispositivos única y exclusivamente a través de estos 4 interfaces de programación. Esto viene a significar que un driver de un dispositivo USB jamás accede directamente al hardware de ese dispositivo, y por otro lado significa que todos los dispositivos USB deben cumplir necesariamente unas especificaciones básicas comunes, ya que deben gestionar adecuadamente los tipos de transferencias que soportan. Adicionalmente, los dispositivos USB se agrupan en clases, de forma que todos los dispositivos de una misma clase cumplen además con unas especificaciones comunes, ya que la clase incide directamente en la manera en que el software interactúa con el dispositivo.

5.2.6. Modelo lógico

Los dispositivos USB pueden tener una o más configuraciones posibles, que definen distintas formas de funcionamiento. A nivel lógico, una determinada configuración es un conjunto de interfaces, donde cada interfaz especifica qué partes del hardware del dispositivo se comunican con el sistema. Cada una de estas partes de hardware se denomina endpoint. Los endpoints son unidireccionales, y se direccionan por un número y por el sentido en que transfieren la información (IN (entrada) si transfieren información hacia el sistema, y OUT (salida) si transfieren información hacia el dispositivo).

La comunicación entre una aplicación y los distintos endpoints de un dispositivo se realiza a través de USB por medio de unos caminos lógicos de transferencias de datos denominados pipes, de forma que cada pipe comunica la aplicación con un determinado endpoint en el dispositivo. Las pipes pueden ser de tipo control (también denominadas de mensaje), que son bidireccionales y con formato especificado por la norma, y de tipo stream, que son unidireccionales (tipo FIFO) y con formato libre no especificado por la norma. Las pipes de control conectan la aplicación con un endpoint de control (formado por una pareja de endpoints uno IN y otro OUT) para realizar transferencias bidireccionales de Control. Las Pipes stream conectan la aplicación con un endpoint para realizar transferencias unidireccionales Isócronas, Interrupción y Bulk.

Todos los dispositivos USB deben implementar los dos endpoints 0 (IN y OUT) para permitir que el sistema establezca la pipe de control por defecto, acceda a la información de identificación y requisitos de configuración y configure el dispositivo. Adicionalmente, USB permite direccionar otros 15 endpoints IN y 15 endpoints OUT por dispositivo FS y otras 2 pipes de control y/o Interrupción por dispositivo LS. Estos endpoints adicionales son opcionales y dependientes de los requisitos de implementación del dispositivo.

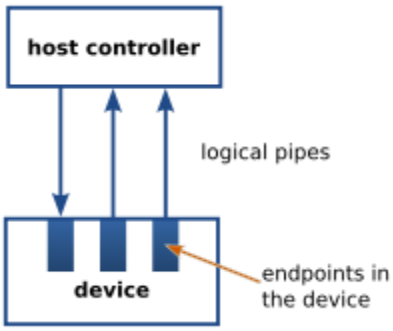


Ilustración 17. Modelo lógico USB [Nec09]

5.2.7. Pasos del funcionamiento

Los pasos que ocurren desde que el usuario introduce el dispositivo USB en el bus hasta que podemos acceder a ese dispositivo son varios. El proceso puede modificarse dependiendo de sus características y puede resultar algo complejo. Procedemos a realizar una breve explicación de dichos pasos con el objeto de que resulten más comprensibles.

1. El sistema arranca con un dispositivo ya conectado.

2. El hub monitoriza los voltajes en cada línea de cada puerto. Cada línea D+, D- está conectada a tierra por una resistencia de 15KΩ. Por el lado del dispositivo esta línea está conectada a 5 voltios igualmente con una resistencia de 1.5KΩ en el lado D- cuando la velocidad es 1.5Mb/s y el lado D+ cuando la velocidad solicitada es 12Mb/s. Al conectar el dispositivo, el hub detecta el aumento de tensión en una de las dos líneas.

3. Cuando el hub detecta un nuevo dispositivo envía un mensaje por el pipe de interrupción al host –es uno de los canales que se ha creado cuando se ha configurado el hub en el bus-. Entonces el host envía al hub un comando para saber por qué puerto ha tenido lugar el evento.

4. Antes de que el hub resetee el dispositivo determina si usa velocidad baja o alta examinando los voltajes en cada línea. Esta información es también devuelta al host cuando este envíe el *Get_Port_Status (Petición de estado del puerto).*

5. Una vez que el host conoce que hay un dispositivo nuevo, envía al hub un mensaje *Set_Port_Feature (establecer características del puerto)* en el que le solicita al hub que resetee el puerto. El reset se realiza poniendo las líneas D+ y D- a tierra durante un mínimo de 10ms (lo normal es que ambas líneas estén en estados contrarios). Solo se resetea el dispositivo solicitado.

6. Para ello hay dos estados especiales de las líneas: chirp J y chirp K. En el primero solo la línea D+ es alimentada y el segundo sólo la línea D- es alimentada. Esto ocurre durante el reset, si se detecta un chirp K, un hub de alta velocidad responde con una secuencia alternada de K y J. Entonces el dispositivo sabe que puede comunicarse con el hub en HS. Si el hub no realiza ninguna acción durante ese tiempo el dispositivo se mantiene en FS. Explicaremos a continuación el proceso de chirp.

7. Una vez indicado al hub que resetee el dispositivo, el host envía continuos *Get_Por_Status* para saber cuándo ha salido el dispositivo del estado de reset. Cuando un dispositivo sale del estado reset se encuentra en un estado por defecto. Sus registros están en un estado de reset y el dispositivo está disponible para realizar transferencias de control sobre el endpoint 0. La dirección lógica por defecto del dispositivo es 00, y el dispositivo es capaz de consumir hasta un máximo de 100 mA del bus.

8. Ahora el host puede enviar mensajes al dispositivo 0 Endpoint 0. Como sólo se configura un dispositivo cada vez, sólo hay un único dispositivo 0 en todo el bus. La respuesta al *Get_descriptor (obtener descriptor)* es un device descriptor con 8 bytes, entre los cuales se encuentra la información de cuál es el tamaño máximo de paquete que se va a transmitir por el pipe. De estos 8 bytes, sólo el primero es interesante, por lo que el reconocimiento se envía tras recibir este byte. A continuación el host solicita al hub que vuelva a resetear el dispositivo.

9. Tras recibir la nueva dirección y almacenarla, el dispositivo pasa a un estado address. A partir de este momento se utiliza la nueva dirección y no la dirección 0. Esta dirección puede ser distinta cada vez que el dispositivo se conecta al bus, aunque se conecte repetidamente en el mismo puerto.

10. En esta ocasión el host está interesado en toda la información. Pedirá primero el descriptor del dispositivo, a continuación uno o varios descriptores de configuración. Cada petición es respondida con el descriptor y toda la información subordinada a ese descriptor. El host limita el número de bytes a recibir en cada petición. Así, en una primera solicitud obtiene el descriptor de configuración exclusivamente, y en una segunda solicitud obtiene lo mismo más los descriptores de interfase y los descriptores de endpoint.

11. Tras conocer todos los detalles del dispositivo busca un driver que gestione las comunicaciones con el dispositivo. Si no encuentra esta información entonces busca drivers que sirvan para el tipo de dispositivo, clase, subclase y tipo de protocolo (todo esto es información extraída del dispositivo). Tras arrancar el driver, este frecuentemente solicita de nuevo al dispositivo toda la información más información específica relacionada con la funcionalidad del mismo.

12. Cuando el dispositivo tiene muchas configuraciones, se dice que es un dispositivo múltiple, primero hay que seleccionar y activar una configuración. El host activa una configuración basándose en información previa disponible, solicitándole al usuario esta información o simplemente activando la primera configuración que encuentre. *Set_configuration (establecer configuración)* insta al dispositivo a que active una configuración y una interfaz. Entonces es cuando se obtiene la información de la interfase necesaria para despertar al driver adecuado.

5.3. Sobre el Chirp Handshake

Durante el proceso de captura de tramas nos podemos encontrar con un pequeño problema debido a una cierta incompatibilidad de velocidades entre el host y los dispositivos. Este suceso es conocido como chirp handshake:

Dado que USB 2.0 promete compatibilidad hacia atrás, no debe haber ningún problema en la comunicación con USB 1.1. Teniendo en cuenta que USB 2.0 utiliza los mismos cables y conectores que USB 1.x, a priori no hay ninguna forma de diferenciar la comunicación a 480 Mbps de la de 12 Mbps. Debido a esto, es necesario notificar a los controladores del host y de los dispositivos que velocidad de transferencia deben proporcionar. Se trata de una nueva necesidad introducida con USB 2.0, ya que bajo USB 1.x se podía diferenciar entre FS y LS basándonos en la posición de la resistencia conectada del lado del dispositivo (**Ilustración 18**).

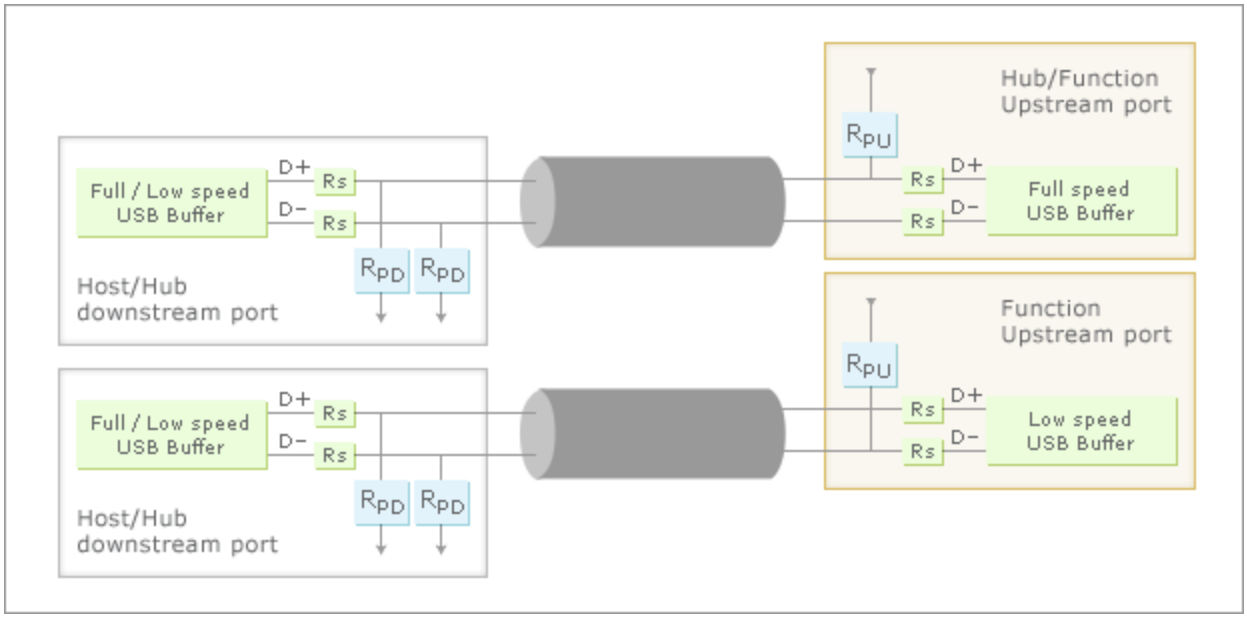


Ilustración 18. Esquema Low/Full speed en USB 2.0 [Nec09]

Debido a que la posición de la resistencia no se puede utilizar para determinar la posición en USB 2.0, necesitaremos otro método para hacerlo. Un dispositivo que soporte HS debe ser capaz de operar también a 12 Mbps. Tomando esta ventaja, al principio todos los dispositivos son conectados como FS y luego los controladores del host y del dispositivo confirman la velocidad de transferencia que soportan. Este proceso es conocido como “chirp handshake” y se completa durante la secuencia de reset en USB. La diferencia entre la comunicación FS y HS se muestra en la **Ilustración 18** y la **Ilustración 19**.

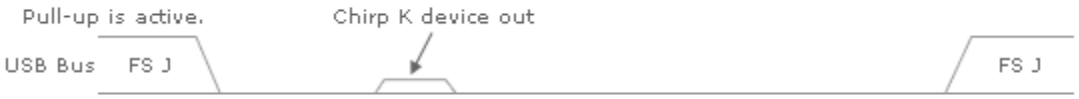


Ilustración 19. Proceso FS

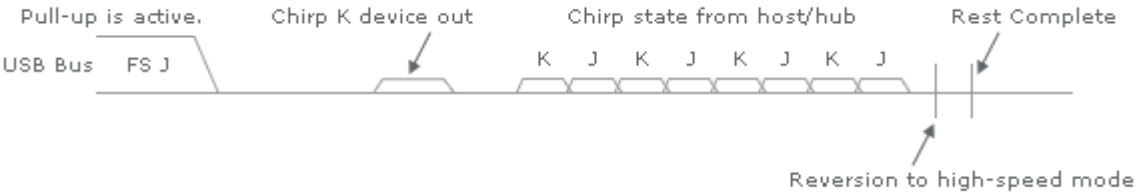


Ilustración 20. Proceso HS

Al final del chirp handshake, la resistencia de pull-up en el dispositivo esta puesta en OFF, permitiendo que esta se comporte como un buffer de USB 2.0 HS. La secuencia de paquetes que se capturan en el analizador en referencia al chirp handshake es como la de la **Ilustración 21**.

Packet	Dir	Chirp J	Time Stamp
979	-->	45.917 µs	00003.7528 0789
Packet	Dir	Chirp K	Time Stamp
980	?	45.683 µs	00003.7528 3543

Ilustración 21. Paquetes con chirping en el analizador

Este suceso nos va a causar ciertos problemas, sobre todo con el dispositivo de reconocimiento de venas de la mano, ya que funcionará a una velocidad diferente a la del analizador. Lo veremos más en detalle en capítulos posteriores.

Capítulo 6:

CATC, analizador USB

En este capítulo hablaremos sobre el analizador que hemos utilizado en nuestro proyecto. Veremos sus características especiales, componentes físicos, opciones de captura y de visualización.

6.1. Descripción

Ilustración 22. Analizador CATC

El analizador que utilizaremos en el proyecto está fabricado por empresa Lecroy. Está provisto de dos interfaces USB en su parte frontal y de otra interfaz USB en su parte posterior. La parte frontal del analizador será la que utilicemos para conectar nuestros dispositivos biométricos que sean objeto de análisis y la parte posterior servirá para conectar el analizador al Host que nos va a servir para monitorizar los resultados, según la **Ilustración 23**. Como podemos observar en la **Ilustración 22**, la carcasa frontal se divide en dos partes:

La parte izquierda está dedicada al proceso de captura de la información y recibe el nombre de USB Analyzer. El dispositivo a analizar se conecta al enchufe USB-A y en el enchufe USB-B conectaremos nuestro PC para poder acceder al software que controle dicho dispositivo.

En la parte derecha se encuentra la zona dedicada a la generación de tramas por parte del analizador, que recibe el nombre de USB Trainer. Esta es una característica que será clave a la hora de realizar los ataques MITM, ya que el hecho de que el analizador pueda generar sus propias tramas nos servirá para suplantar a la postre los dispositivos USB previamente analizados. El conexionado de cables para este propósito comprende la conexión de la parte Trainer (USB-B) con la parte Analyzer (USB-A), de forma que parezca que hay conectado un dispositivo al analizador, y la conexión de la parte Analyzer con el PC (al igual que antes, para controlar el software del dispositivo).

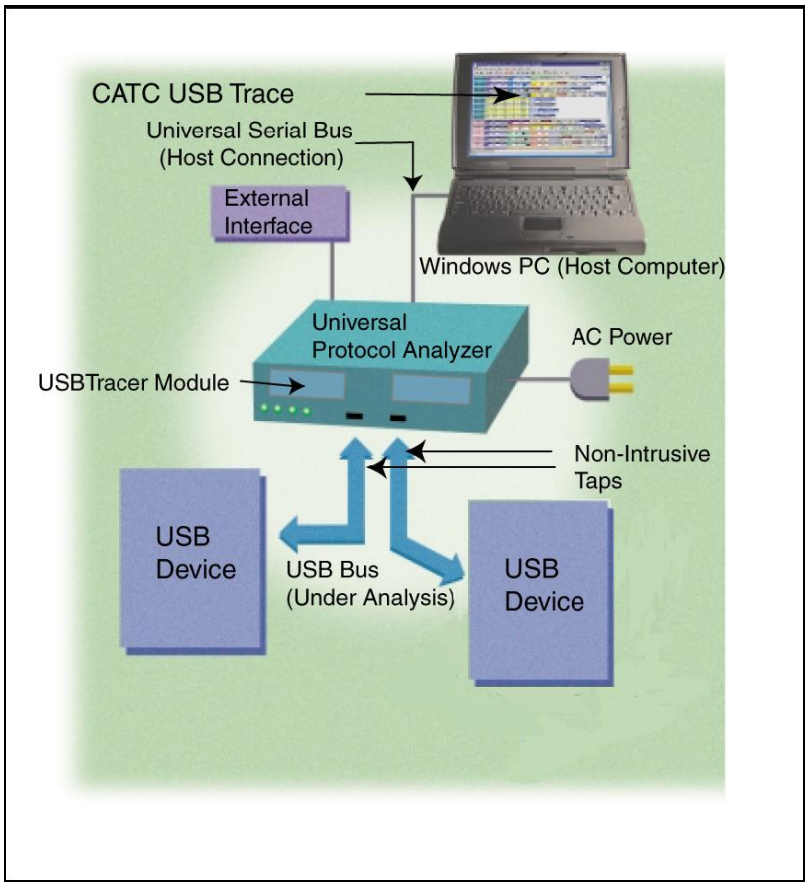


Ilustración 23. Conexionado habitual del analizador

6.2. Características principales

En este apartado hablaremos de las características principales del analizador, tanto a nivel hardware como a nivel software, dando una idea global de su funcionamiento.

6.2.1. General

- Total compatibilidad con la especificación USB 2.0.
- Hardware reconfigurable para modificaciones futuras del fabricante.
- Firmware actualizable también según mejoras del fabricante.
- Soporte para todas las velocidades de USB (480 Mbps, 12 Mbps y 1.5 Mbps).
- Captura en 2 canales simultáneamente y a múltiples velocidades.
- Posibilidad de capturar y generar tramas al mismo tiempo.

6.2.2. Componentes físicos

- Instalación USB PnP (sin necesidad de configuración).
- 512 MB de memoria física.
- 2 canales de captura en cada una de las dos partes frontales compatibles con LS, FS o HS, como vemos en la **Ilustración 24** (Channel 0 y Channel 1).
- Velocidad FS entre el PC y el analizador para monitorización.

Ilustración 24. Carcasa del analizador

6.2.3. Opciones de captura

- Visión y enumeración de todas las tramas USB con posibilidad de despliegue de todas ellas hasta el nivel de bit.
- Posibilidad de realizar triggering durante la captura, con objeto de detener el proceso cuando se produzca un evento determinado (finalización de la transmisión de datos, tamaño de la captura superior al deseado, etc.).
- Filtros para evitar visualizar transacciones intrascendentes (por ejemplo SOF's que se repiten con asiduidad y no revelan información importante)
- Filtro de captura de tráfico en tiempo real.
- Buffer de captura de tamaño ajustable desde 0.4 MB a 512 MB

6.2.4. Opciones de visualización

- Visualización gráfica de paquetes del bus, transacciones y transferencias.
- Visualización de las estadísticas de las capturas.
- Posibilidad de colocar marcas para visualizar mejor la comunicación (por ejemplo para remarcar en qué momento comienza el proceso de setup).
- Posibilidad de ocultar tramas no relevantes y numerosas que dificultan el visionado de la comunicación.

Capítulo 7:

Dispositivos biométricos utilizados

En este capítulo veremos los dispositivos biométricos que vamos a utilizar para el proyecto. Para llevar a cabo el análisis de los diferentes dispositivos utilizados, hablaremos de sus características principales, del proceso de identificación que hemos seguido en cada uno, de la interfaz que nos ofrece el software y de las herramientas de las que hemos dispuesto para realizar las pruebas.

7.1. Panasonic Authenticam PrivateID

Ilustración 25. Panasonic Authenticam [Pan09]

7.1.1. Características principales

La cámara de Panasonic tiene como características principales su capacidad de tomar imágenes del iris y la capacidad de grabar video. Esta segunda opción la obviaremos dada su irrelevancia en nuestros propósitos. Tiene un tamaño y un peso pequeños. En las especificaciones técnicas no nos indican nada acerca de algoritmos de seguridad que se apliquen a la comunicación o a los datos.

7.1.2. Proceso de identificación seguido

El proceso de identificación se realiza colocando el ojo a unos 20 cm de la cámara y mirando hacia una luz roja que procede de esta. Cuando el software de la cámara enfoca el ojo en una buena posición, realiza la captura de la imagen.

7.1.3. Interfaz de uso

Para la interfaz de uso del dispositivo, hemos utilizado dos tipos de software diferentes:

- Por una parte, hemos utilizado el software de Panasonic que viene incluido con la cámara. La interfaz de uso que nos ofrece nos ha servido para realizar las primeras pruebas y comprobar cómo se transmite la información entre el dispositivo y el analizador. Se trata de una aplicación que toma una imagen del ojo, pero no la almacena ni ofrece autenticación.

- Por otra parte, hemos utilizado un programa diseñado en el GUTI (Grupo de Investigación de Tecnologías de Identificación) (**Ilustración 26**) que realiza la captura de forma idéntica a como se realiza con el software incluido con la cámara. Además incluye la posibilidad de registrarse, de autenticarse y almacena las imágenes obtenidas.

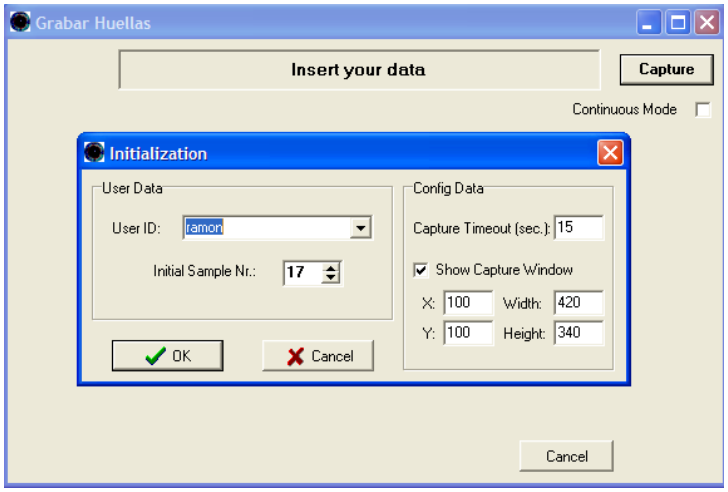


Ilustración 26. Aplicación desarrollada en el grupo de investigación

7.1.4. Herramientas utilizadas

Las herramientas utilizadas han sido por una parte los drivers de la cámara que hemos podido obtener de su página web. Por otro lado, hemos utilizado el software de Panasonic (incluido con la cámara) que nos ha servido para familiarizarnos con ella.

Por último, hemos utilizado un programa diseñado en el laboratorio que nos ha permitido obtener la información necesaria para realizar los ataques pertinentes, ya que como hemos dicho, permite obtener las imágenes realizadas y realizar autenticación de usuarios

7.2 DELL Biometric Coprocessor

Ilustración 27. Dispositivo de huella DELL [Del09]

7.2.1. Características principales

El dispositivo de huella de DELL tiene un modo de identificación por barrido. Según las especificaciones técnicas, la comunicación con este dispositivo estará cifrada con el algoritmo RSA.

7.2.2. Proceso de identificación seguido

El software del fabricante permite acceder al dispositivo y proceder a una autenticación por usuario y por diferentes dedos. El proceso se realiza haciendo un barrido del dedo (cualquier dedo en la autenticación simple, el que se requiera en la autenticación por software) a través del receptor.

7.2.3. Interfaz de uso

La interfaz de uso del software que nos ofrece DELL se compone de un menú donde se nos ofrece autenticarnos o registrarnos. Para nuestro proyecto utilizaremos cualquiera de estos métodos para la captura de información y la suplantación.

7.2.4. Herramientas utilizadas

Las herramientas utilizadas han sido por una parte los drivers que hemos podido obtener en la página web de DELL y por otro lado el software que viene incluido con el

dispositivo, ya que como hemos dicho, permite realizar autenticación y nos será suficiente para nuestro proyecto

7.3. Biometrika FX-3000 Fingerprint Scanner

Ilustración 28. Dispositivo Biometrika de huella dactilar [Bio09]

7.3.1. Características principales

El dispositivo de huella de Biometrika tiene una superficie de detección por huella completa (sólo hay que apoyar el dedo, no hay que hacer un barrido). También viene incorporado con una ranura para tarjetas inteligentes que no utilizaremos en el proyecto. En las especificaciones técnicas se detalla que se utilizan algoritmos simétricos de 128 bits junto con claves no repetibles. Se incluye también la posibilidad de enviar la información del dispositivo al ordenador con diferentes niveles de seguridad. Esto va a dar lugar a que podamos obtener imágenes en claro como la de la **Ilustración 29**.

Ilustración 29. Huella capturada con el dispositivo de Biometrika

7.3.2. Proceso de identificación seguido

La identificación la realizaremos mediante el software que proporciona el fabricante. Incluye un proceso de configuración con objeto de que las huellas recogidas estén bien centradas y no haya errores. El software incorpora un proceso de autenticación y otro de reconocimiento.

7.3.3. Interfaz de uso

La interfaz de uso de este dispositivo está formada por un programa que permite realizar registro y autenticación de usuarios. En cada proceso, se realizan 3 capturas por cada huella y en cada una se nos indica cómo de buena ha sido la captura de la huella con un porcentaje.

7.3.4. Herramientas utilizadas

Hemos utilizado los drivers y el programa que nos ofrece biométrica. No hemos requerido herramientas extra para realizar nuestros ataques.

7.4. Fujitsu PalmSecure

Ilustración 30. Dispositivo de reconocimiento de venas Fujitsu [Fuj09]

7.4.1. Características principales

El dispositivo de reconocimiento por venas de la mano tiene unas dimensiones reducidas. Incluye un reposamuñecas para realizar el reconocimiento. En las especificaciones nos indican que este dispositivo cifra las imágenes con el algoritmo AES de 256 bits.

7.4.2. Proceso de identificación seguido

La identificación la realizaremos mediante el software que proporciona el fabricante. Este dispositivo, como el de Biometrika, ofrece la posibilidad de autenticación o de identificación. El proceso se completa tras dos capturas de la imagen de las venas de la mano mediante infrarrojos.

7.4.3. Interfaz de uso

La interfaz de uso de este dispositivo comprende un programa (incluido con el dispositivo) que permite realizar identificación y autenticación. Después de ello, se nos muestra por un instante la imagen capturada y se nos indica si el proceso ha sido exitoso.

7.4.4. Herramientas utilizadas

Como en casi todos los casos anteriores, nos ha sido suficiente con los drivers del fabricante y con el software del que dispone el dispositivo.

Capítulo 8:

Ataques realizados y resultados obtenidos

En este capítulo mostraremos todas las pruebas realizadas durante el proyecto, las documentaremos y comentaremos. También hablaremos de los problemas surgidos durante los ataques y mostraremos cuando sea menester las trazas que ha capturado el analizador. A su vez, explicaremos los resultados obtenidos y los contrastaremos de forma que lleguemos a una conclusión de la seguridad que ofrecen estos dispositivos frente a los ataques realizados.

8.1 Captura de la información

Debido a la naturaleza diferente de cada dispositivo USB utilizado, las características de la comunicación entre ellos también serán diferentes. Después de realizar un análisis exhaustivo de cada uno de ellos, procederemos con una pequeña descripción de los resultados obtenidos en el proceso de captura de la información por parte del analizador.

El conexionado dispositivo-Host durante el proceso de captura será idéntico para todos los dispositivos, siendo conectados estos al puerto USB A y siendo conectado el Host (el ordenador) al puerto USB B, tal y como vemos en la **Ilustración 31**. De este modo se observa perfectamente el papel de cada componente en el proceso de ataque MITM, siendo el analizador un espía en la comunicación.

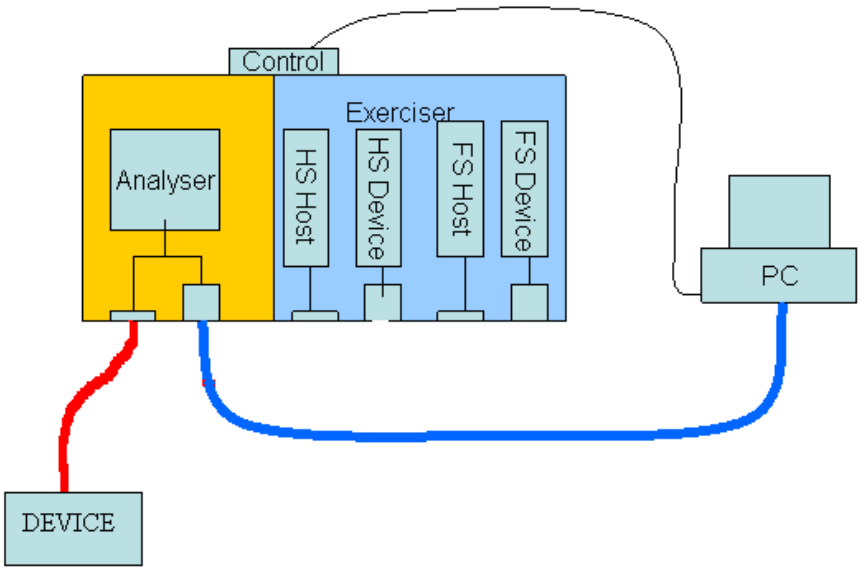


Ilustración 31. Conexionado durante el proceso de captura

Previamente a la captura de la información realizada en los dispositivos biométricos, hemos realizado el proceso con un dispositivo de almacenamiento USB (lápiz o llave USB). Este experimento nos servirá para reconocer el tráfico USB y observar el comportamiento del analizador.

8.1.1. Dispositivo de almacenamiento USB

Ilustración 32. Dispositivo USB analizado

Este será la única operación que realicemos sobre este dispositivo, puesto que solo queremos examinar cómo se realiza la comunicación USB. El proceso comenzará poniendo el analizador a capturar información estando el lápiz USB sin conectar, con objeto de poder observar el intercambio de información a la hora de reconocer el dispositivo. Cuando conectamos el dispositivo, se comienzan a monitorizar los datos obtenidos de la comunicación dispositivo-Host.

Transfer	H	Control	ADDR	ENDP	bRequest	wValue	wIndex		Time	Time Stamp
31	S	SET	2	0	CLEAR_FEATURE	ENDPOINT_HALT	For Endpoint # 0x00		125.783 µs	00007.5635 5445
Transfer	H	Control	ADDR	ENDP	bRequest	wValue	wIndex	STALL	Time	Time Stamp
32	S	GET	2	0	0xFE	0x0000	0x0000	0x08	125.300 µs	00007.5636 5492
Transfer	H	Control	ADDR	ENDP	bRequest	wValue	wIndex		Time	Time Stamp
33	S	SET	2	0	CLEAR_FEATURE	ENDPOINT_HALT	For Endpoint # 0x00		177.667 µs	00007.5637 5510
Transfer	H	Interrupt	ADDR	ENDP	Bytes Transferred	Time	Time Stamp			
34	S	OUT	2	1	31	368.033 µs	00007.5639 1170			
Transfer	H	Bulk	ADDR	ENDP	Bytes Transferred	Time	Time Stamp			
35	S	IN	2	2	36	202.450 µs	00007.5642 0752			

Ilustración 33. Captura CATC llave USB

Como podemos apreciar en la **Ilustración 33**, en primer lugar se produce el intercambio de información de configuración. Después del proceso de control se envían paquetes de tipo SOF (Start Of Frame) que se envían al bus al principio de cada trama, permitiendo a los dispositivos isócronos sincronizarse con el bus. Después de este paso procedemos a intercambiar información entre el Host y el dispositivo (transferimos un fichero) dando lugar a un volcado de datos y envío de paquetes ACK (acuse de recibo) y NACK (acuse de recibo negado). La dirección de estos paquetes dependerá de quién sea emisor y de quién sea receptor.

Después de la captura observamos que disponemos de toda la información intercambiada, por lo que si utilizamos la opción del analizador “Export - Data”, tendremos en un fichero de texto los datos que se han enviado en hexadecimal, como en la **Ilustración 34**. En esta ocasión no nos interesa obtener esta información, pero después comprobaremos que puede ser muy valiosa.

```
File \\owner-efbde0f78\ramón\trazas usb\llave usb\pruebaLlave4.usb.
Transfer from 21 to 47.
Export   Data for Address 001 All Endpoints (Direction:OUT)
Data @ Transfer #21, 31 bytes Out.
55 53 42 43 C8 5B 0A 8A 00 04 00 00 00 00 0A 2A 00 00 00 06 27 00 00 02 00 00
00 00 00 00 00

Data @ Transfer #22, 1024 bytes Out.
50 61 63 6B 65 74 2C 43 68 2C 54 69 6D 65 20 53 74 61 6D 70 2C 48 2C 44 72 2C
53 70 2C 53 79 6E 63 2C 50 69 64 2C 46 72 61 6D 65 2C 41 64 2C 45 70 2C 43 52
43 35 2C 43 52 43 31 36 2C 45 4F 50 2C 48 69 45 4F 50 2C 52 65 73 65 74 2C 53
75 73 2C 52 73 6D 2C 43 68 4B 2C 43 68 4A 2C 46 53 4B 2C 46 53 4A 2C 53 45 30
2C 53 45 31 2C 56 42 75 73 2C 53 43 2C 48 62 41 2C 50 2C 53 2C 45 2C 45 54 2C
45 78 74 65 72 6E 61 6C 2C 50 4C 2C 44 61 74 61 0D 0A 32 30 30 36 32 2C 2C 30
30 30 30 31 2E 37 30 31 32 20 33 35 31 39 2C 2C 4F 2C 48 2C 2C 30 78 38 37 2C
```

Ilustración 34. Porción de datos capturados

Con esta comunicación nos damos cuenta, a grandes rasgos, de cómo funciona el protocolo USB y nos familiarizamos con el formato de visualización del analizador. Con la visualización de los datos obtenidos damos por concluido el proceso de captura de información Host – Llave USB.

8.1.2. Panasonic Authenticam PrivateID

Con la cámara de Panasonic hemos seguido los mismos pasos que con el dispositivo de almacenamiento USB, pero hemos obtenido resultados diferentes. Describimos a continuación los pasos que hemos dado en este ataque.

En primer lugar, hemos comenzado a capturar antes de conectar la cámara al analizador, como hemos hecho antes, con el fin de observar el proceso de setup. Después hemos conectado la cámara y hemos hecho una foto de nuestro iris. El proceso de captura termina deteniendo el analizador.

El proceso de comunicación entre el Host y la cámara engloba transferencias de Control e Isócronas. Las transferencias de control informarán de las características del dispositivo y de la comunicación, a la par que servirán de herramienta “*keep alive*”. El analizador proporciona una herramienta muy útil para observar el proceso global llamada Navigator (vemos un ejemplo en la **Ilustración 35**). Con ella podemos ver en un simple esquema, y diferenciados por colores, todos los procesos que se llevan a cabo.

En el caso de la cámara de iris, el proceso es de esta forma:

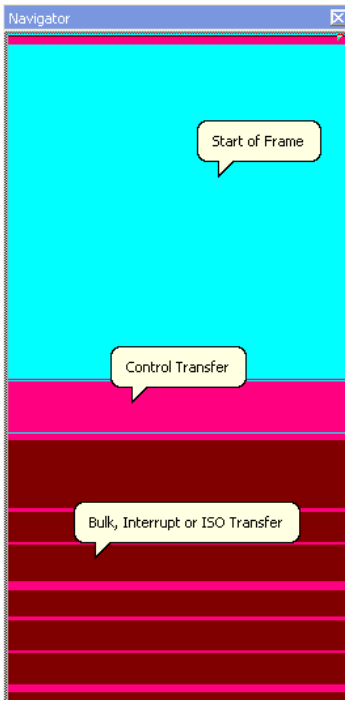


Ilustración 35. Proceso Navigator Authenticam

Aquí podemos observar el proceso inicial de configuración a través de una captura del analizador. Vemos como, en primer lugar, el dispositivo proporciona su Descriptor donde informará de sus diferentes características como la ID del producto o su número de serie. Posteriormente le será asignada una dirección para que la comunicación se realice a través de ella (*set address –establecer dirección-*) y el dispositivo proporcionará un descriptor con su configuración (número de interfaces, máximo voltaje soportado, etc.)

Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
GET	0	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE descriptor	4.032 ms	00006.2576 5088

Control	ADDR	ENDP	bRequest	wValue	Time	Time Stamp
SET	0	0	SET_ADDRESS	New address 1	61.998 ms	00006.3080 4977

Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
GET	1	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE descriptor	4.000 ms	00006.3576 4878

Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
GET	1	0	GET_DESCRIPTOR	CONFIGURATION type	0x0000	CONFIGURATION descriptor	4.000 ms	00006.3608 4866

Ilustración 36. Proceso de configuración

El analizador nos ofrece un detalle de observación hasta el nivel de paquetes, por lo que desplegando los niveles podemos llegar a tratar directamente con los bits enviados. Según lo visto antes en el apartado de “*transacciones de control*”, verificamos que la comunicación es la prevista:

1. Transacción de configuración (SETUP)
2. Transacción de datos (IN / OUT)
3. Transacción de estado (ACK)

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time Stamp
0	S	GET	0	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE descriptor	00006.2576 5088

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
0	S	0xB4	0	0	0	D->H	S	D	GET_DESCRIPTOR	DEVICE type	0x0000	64	0x4B	00006.2576 5088

Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
63	-->	S	00000001	0xB4	0	0	0x08	233.330 ns	183.320 ns	00006.2576 5088

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
64	-->	S	00000001	0xC3	8 bytes	0xBB29	233.330 ns	283.330 ns	00006.2576 5273

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
65	<--	S	00000001	0x4B	250.000 ns	988.283 µs	00006.2576 5784

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
1	S	0x96	0	0	1	18 bytes	0x4B	2.000 ms	00006.2584 5081

Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
2	S	0x87	0	0	1	0 bytes	0x4B	1.032 ms	00006.2600 5078

Ilustración 37. Transacciones de control

El proceso de transferencia isócrona se encuentra inmerso entre trasferencias de control. Se trata de una transferencia que engloba un número importante de transacciones que envían datos (que serán la información del iris captada por la cámara).

Podemos observar un tamaño de paquete máximo de 960 bytes y también el hecho de que el Endpoint en el que se realiza la operación es diferente.

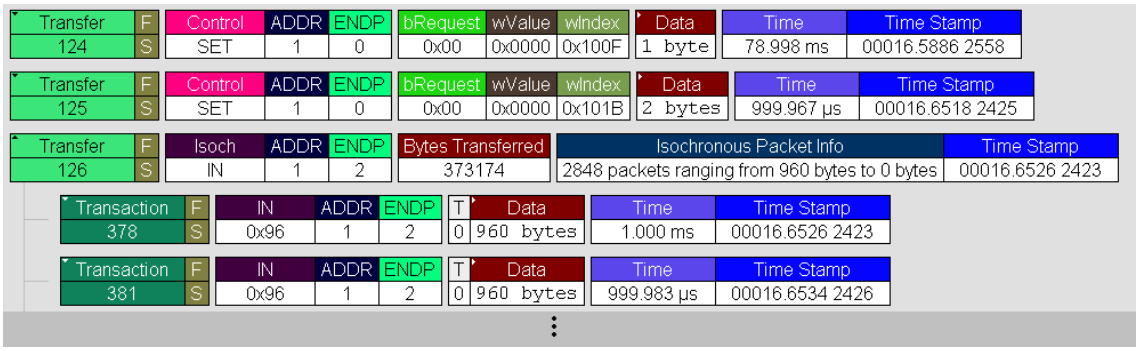


Ilustración 38. Transacciones de datos

Será en este punto cuando se envíe la imagen capturada del iris al Host. Si procedemos igual que en el apartado anterior y movemos los datos a un fichero de texto, tendremos la imagen capturada en hexadecimal. Si pasamos los datos “en crudo” (formato .raw) a cualquier programa de tratamiento de imagen, tendremos aparentemente ruido (**Ilustración 39**).

Ilustración 39. Imagen "en crudo"

Cuando el software de tratamiento de imagen nos pida la resolución de la imagen (en el formato .raw no está fijada), fijaremos la que ya conocíamos, puesto que previamente hemos obtenido imágenes de prueba y todas las capturas tendrán el mismo tamaño. Con los datos en crudo colocados a la resolución adecuada en píxeles, obtendremos la imagen del iris de forma nítida (**Ilustración 41**).

Sabemos que dicha imagen tiene un formato .JPEG puesto que teníamos imágenes previas y además en los datos en crudo se puede apreciar una cabecera como la de la **Ilustración 40** que corresponde con .JPEG. El formato JFIF (Formato de Intercambio de Archivos JPEG) que se ve en la ilustración es el más elemental y sólo contiene la imagen. Este formato es el que suelen tener las imágenes con extensión JPEG.

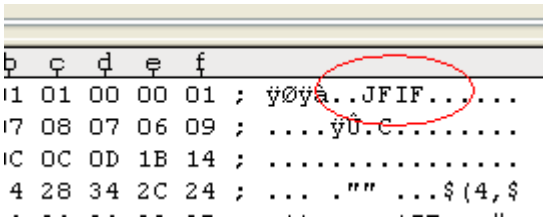


Ilustración 40. Ejemplo de imagen en formato JPEG/JFIF [Jpg07]

La diferencia entre la imagen del ruido (**Ilustración 39**) y la imagen nítida (**Ilustración 41**), viene dada únicamente por el hecho de haber introducido la resolución adecuada en el segundo caso, ya que de otra forma los píxeles no coinciden con la foto original y una desviación de unos pocos da como resultado una imagen totalmente diferente. De este modo completamos el proceso de captura con la cámara de forma exitosa, finalizando con ello el primer ataque. En este caso nos será indiferente realizar la captura con el software de Panasonic que con el que tenemos en el laboratorio, ya que no es necesario ningún proceso de autenticación.

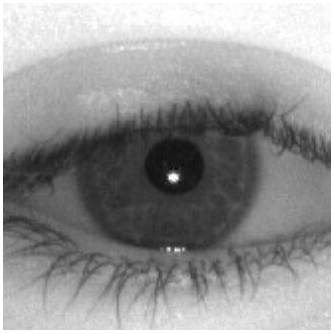


Ilustración 41. Imagen definida

8.1.3. DELL Biometric Coprocessor

Los pasos realizados con el dispositivo de huella dactilar de DELL para completar el proceso de captura han sido los siguientes:

En primer lugar, y como venimos haciendo hasta ahora, hemos puesto el analizador a capturar previamente a conectarle el dispositivo. Hemos arrancado el software de DELL y nos hemos autenticado de forma exitosa. Luego hemos detenido la captura. Pasamos a comentar los resultados del analizador.

En este caso, las transferencias serán de tipo Control y de tipo Bulk, ya que el envío de información será mayor. Con la opción Navigator del analizador podemos ver un esquema del conjunto de las operaciones (**Ilustración 42**).

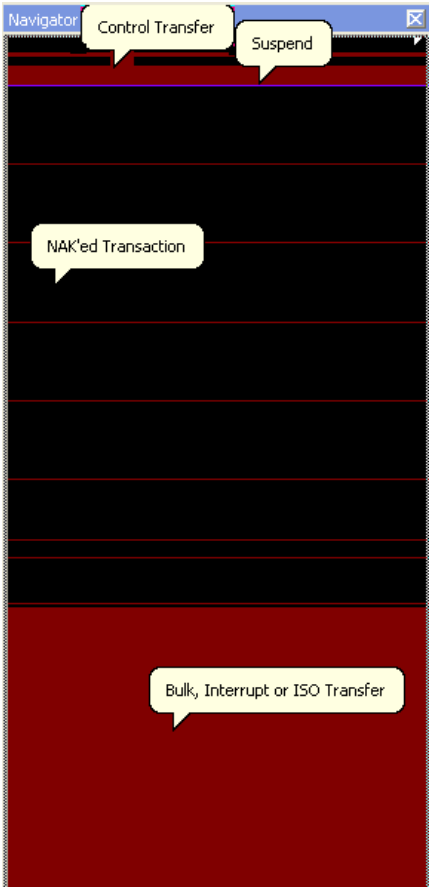


Ilustración 42. Proceso Navigator DELL

Seguidamente al proceso de control, se realiza en la comunicación el volcado de datos “bulk”. Se trata de una larga sucesión de envíos de transferencias Bulk-IN/OUT donde se transmitirá la información que se obtenga del análisis de la huella dactilar. El proceso es como el siguiente:

Transfer 102	F S	Bulk	ADDR	ENDP	Bytes Transferred	Time Stamp											
		IN	1	1	64	00020.7208 1300											
Transaction 43945	F S	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp								
		0x96	1	1	1	64 bytes	0x4B	27.999 ms	00020.7208 1300								
Transfer 103	F S	Bulk	ADDR	ENDP	Bytes Transferred	Time Stamp											
		OUT	1	2	9	00020.7432 1251											
Transaction 43946	F S	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp								
		0x87	1	2	1	9 bytes	0x4B	699.156 ms	00020.7432 1251								

Ilustración 43. Volcado de datos

En este caso, también podemos diferenciar los Endpoints de los procesos Bulk-IN y Bulk-OUT. El proceso de volcado se repite de forma continua hasta llegar a un punto en el cual se suspende la comunicación y se vuelve a retomar. Este punto será de gran trascendencia cuando evaluemos el producto y veamos sus sistemas de seguridad:

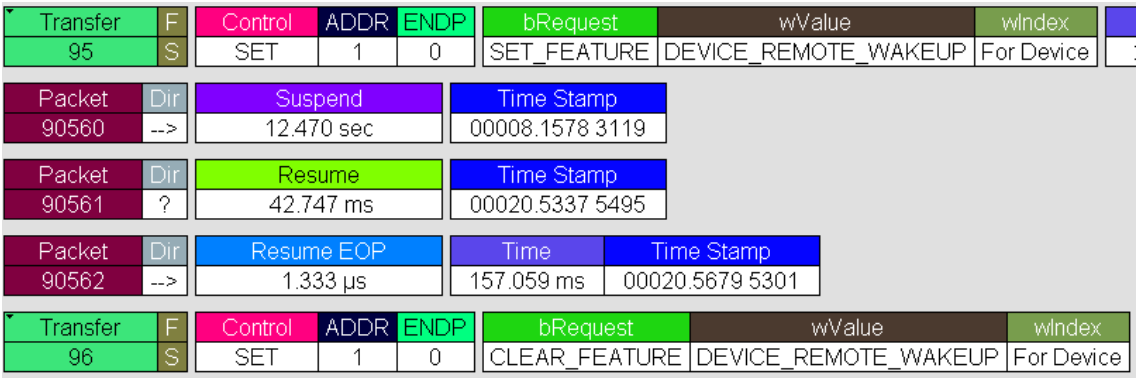


Ilustración 44. Punto de suspensión

Volvemos a llevar a cabo el paso de recoger los datos capturados e intentar formar una imagen con ellos que sea nuestra huella dactilar. En este caso, el intento será infructuoso, ya que el lector de huella de DELL cifra las imágenes que envía al Host mediante el algoritmo RSA. La resolución en píxeles también nos es desconocida puesto que no hemos conseguido ninguna imagen de huella del dispositivo de DELL, pero aun conociéndola, no llegaríamos a obtener la imagen en ningún caso. El resultado obtenido ha sido el de la **Ilustración 45**.

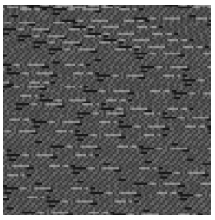


Ilustración 45. Imagen cifrada DELL

8.1.4. Biometrika FX-3000

Repetimos los pasos para capturar la información enviada entre el Host y el dispositivo de huella de Biometrika. En la comunicación con este dispositivo encontramos bastantes similitudes con el dispositivo de DELL, pues ambos realizan procesos de control y volcado de datos. La diferencia más significativa es que con este dispositivo la transferencia de datos se realiza de forma discontinua con ráfagas de tramas NACK de por medio. Vamos por partes:

En primer lugar, conectamos el dispositivo al analizador una vez ha comenzado a capturar. Llevamos a cabo la autenticación con el software de forma exitosa y terminamos la comunicación. Con este lector de huella tenemos la posibilidad de aplicar seguridad o no a la comunicación. Ya que estamos analizando la seguridad del dispositivo, vamos a realizar las operaciones de captura con todas las posibilidades que nos ofrece el software, aunque realmente, no habrá ninguna diferencia entre un caso u otro hasta que no comencemos a procesar la imagen obtenida, pues la comunicación es idéntica. El resultado de la comunicación, desde el punto de vista del Navigator, es como el de la **Ilustración 46**.

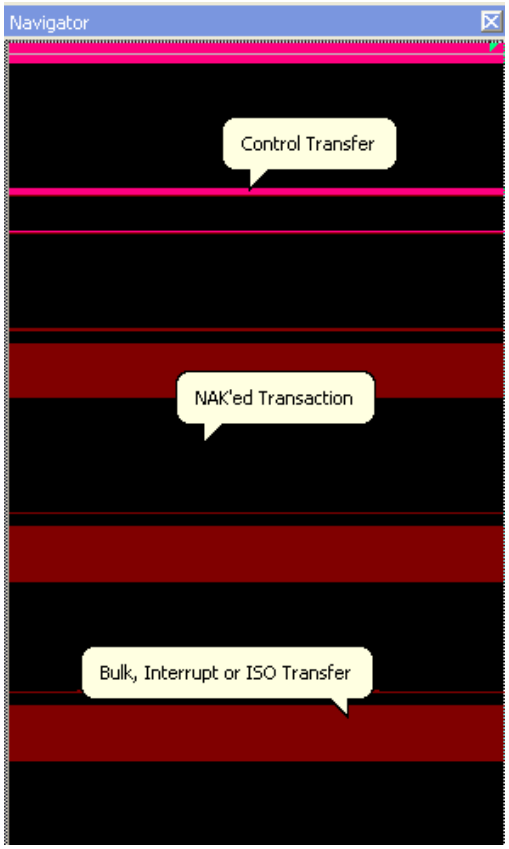


Ilustración 46. Proceso Navigator Biometrika

El proceso de control se completa como ya hemos visto, asignando una dirección al dispositivo y proporcionando éste los descriptores requeridos. Posteriormente se produce el proceso de volcado de datos, que como vemos en la imagen superior, aparece separado por NACK’s, debido a que el Host no puede aceptar todos los datos de la transmisión de forma continua debido a los tiempos de procesamiento.

Llegados a este punto, es cuando podemos hacer una diferenciación entre los grados de seguridad marcados en un principio para el dispositivo, como vemos en la **Ilustración 47**. Hemos hecho tres capturas para tres niveles de seguridad diferentes (sin seguridad, seguridad media y alta seguridad).

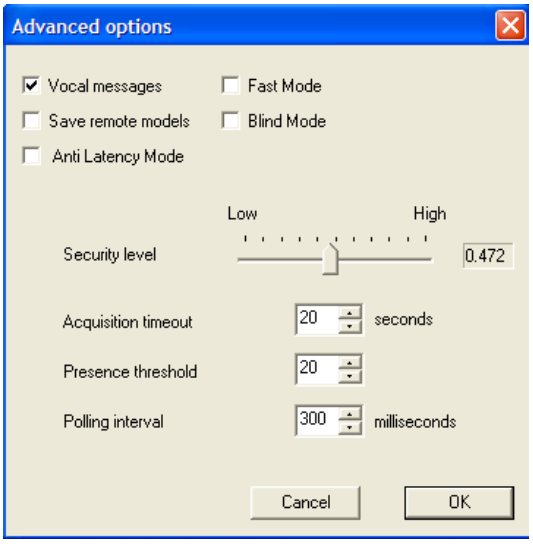


Ilustración 47. Opciones de seguridad Biometrika

Esto se traduce en un cifrado de los datos enviados en la comunicación diferente en cada caso. Después de obtener los datos del analizador y llevarlos al software de tratamiento de imagen, nos encontramos con dos resultados diferentes.

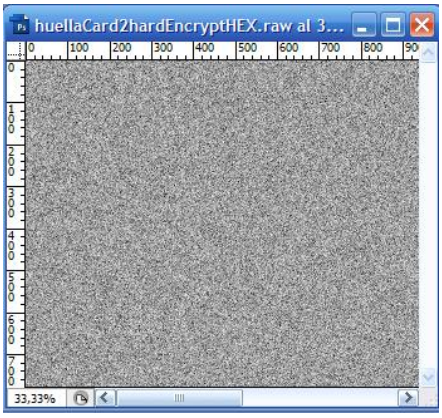


Ilustración 48. Nivel de media o alta seguridad con Biometrika

Ilustración 49. Nivel de seguridad nula

Como podemos apreciar en las imágenes, con el nivel de seguridad nula la imagen se transfiere en claro (**Ilustración 49**), mientras que si el nivel de seguridad sube algún grado, la imagen ya se transfiere cifrada y lo único que obtendremos será ruido (**Ilustración 48**). En este caso, hemos obtenido la imagen nítida llevando a cabo el proceso de tomar los datos en crudo y pasarlos al software de tratamiento de imagen. No obstante, hay una diferencia respecto al caso de la cámara de Panasonic y es que del dispositivo de Biometrika no habíamos obtenido imágenes previamente, por lo que a la hora de decidir la resolución en píxeles el proceso no es tan sencillo.

Lo que hemos hecho ha sido, en primer lugar, colocar los datos en crudo en el programa de tratamiento de imagen a una resolución aleatoria (el software nos pide la resolución para presentar la imagen). Posteriormente, nos hemos fijado en ciertos puntos que se repetían de forma periódica en una misma línea (esto es debido a que los píxeles contiguos en una imagen se asemejan) y hemos hecho un cálculo (ayudados por el programa) del número de píxeles que había entre dos de los puntos (**Ilustración 50**). Este número será el ancho de la imagen transmitida. Hallaremos el alto de la misma variando los valores que pasemos al programa hasta que la imagen sea nítida.

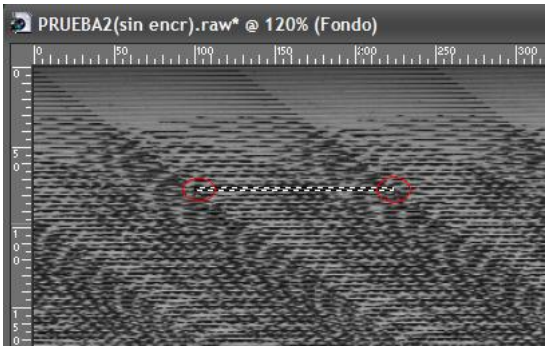


Ilustración 50. Hallar la resolución de la imagen por puntos semejantes

Por lo tanto, de entre todos los intentos realizados, únicamente hemos conseguido una imagen poniendo el nivel de seguridad a cero.

8.1.5. Fujitsu PalmSecure

En este caso procedemos a capturar el tráfico que se produce entre el dispositivo de reconocimiento de venas de la mano y el analizador. Para llevarlo a cabo seguimos los siguientes pasos:

Ponemos el analizador a capturar tráfico y posteriormente conectamos el dispositivo. Realizamos la autenticación con el software de Fujitsu, el cual requiere dos capturas de imágenes de la palma de la mano con éxito y damos por concluida la comunicación.

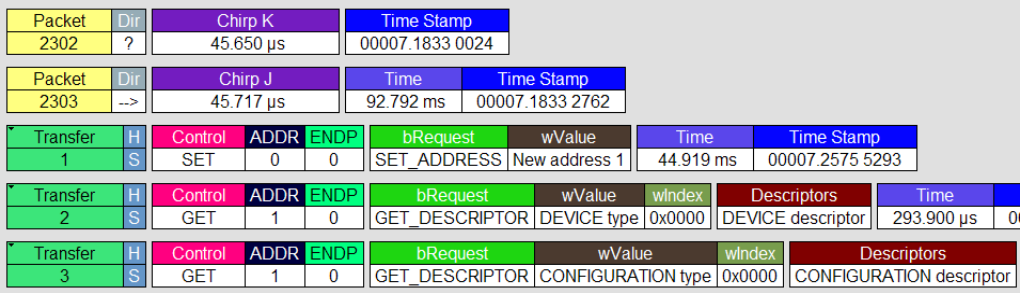


Ilustración 51. Proceso de chirp previo a la configuración

En el caso del dispositivo de reconocimiento de venas encontramos una pequeña diferencia respecto a los demás dispositivos. Ésta es producida por una diferencia de velocidades entre el Host y el dispositivo (ya que en este caso se utiliza HS, modo USB 2.0), produciéndose un envío de tramas conocidas como “*chirp handshake*” previamente al proceso de configuración del dispositivo (**Ilustración 51**).

El resto de las operaciones son de control y de volcado, muy similares a las de los dispositivos de huella dactilar analizados pero con bastantes más transferencias de control. Procedemos, como en el resto de casos, a extraer únicamente los datos que el dispositivo envía al Host (las imágenes de las palmas de la mano).

Tras copiar los datos en crudo al software de tratamiento de imágenes y obtener la imagen que se transfiere del dispositivo al host, nos encontramos con un caso curioso. La imagen que se transmite está dividida en varias partes. En la parte superior están las luces de los infrarrojos, en la parte media está la imagen de la palma de la mano con ruido entremedias y en la parte inferior se encuentra otra imagen de los dedos.

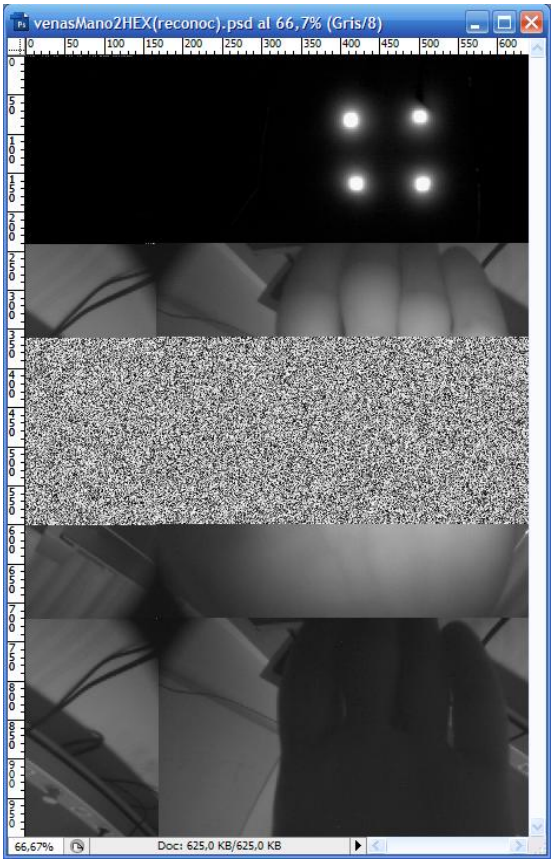


Ilustración 52. Imagen obtenida en la captura

La parte valiosa de la imagen (las venas de la mano) se encuentra cifrada y no podemos hacer nada con ello (**Ilustración 52**). La imagen se podría llegar a visualizar basándonos en que los puntos contiguos de la mano son muy semejantes y se podría hacer una correlación justo en el límite de la imagen con el ruido, pero esto es ajeno a los propósitos de nuestro proyecto.

Con el dispositivo de venas de la mano hemos concluido nuestro proceso de captura de información, que es nuestro primer ataque. Hemos conseguido, no solo colocarnos en medio de la comunicación y monitorizarla, sino que también hemos obtenido los datos enviados entre ambas partes. Todo ello da lugar a dos conclusiones:

- Podemos conseguir información personal que debería ser secreta.

- Conocemos el modo en el que actúan ambas partes, por lo que podemos hacernos pasar por una de ellas, llegando incluso a suplantar la identidad de una persona.

8.2 Creación y Envío de trazas Dispositivo-Host

Una vez concluida la fase de captura de información entre el dispositivo y el host, nos disponemos a finalizar con el proceso de ataques MITM. Después de recopilar información sobre los dispositivos, examinarlos y visualizar la información capturada por el analizador, ya estamos en posición de realizar ataques por suplantación de identidad.

En este caso, el analizador constituye una potente herramienta para desarrollar nuestros ataques, ya que permite el envío de tramas en dirección analizador – host, de forma que, habiendo creado tramas del estilo de las que hemos capturado, podemos enviarlas haciéndonos pasar por el dispositivo utilizado. El analizador permite crear una especie de plantillas llamadas “archivos .utg” (ejemplo en la **Ilustración 53**), en las que se describe el comportamiento que debe tener el dispositivo en forma de script modificable, ante las peticiones del host y viceversa. También contempla el modo de envío de tramas host – analizador con el fin de suplantar al host, pero no será utilizado en nuestro proyecto.

Por lo tanto, haciendo que el analizador genere archivos que se hagan pasar por un determinado dispositivo y haciendo que se genere tráfico entre el analizador y el host podremos suplantar a ese dispositivo. Siempre y cuando no esté provisto de sistemas de seguridad que dificulten nuestra tarea. A continuación, detallaremos los pasos que hemos realizado para llevar a cabo los ataques en cada dispositivo.

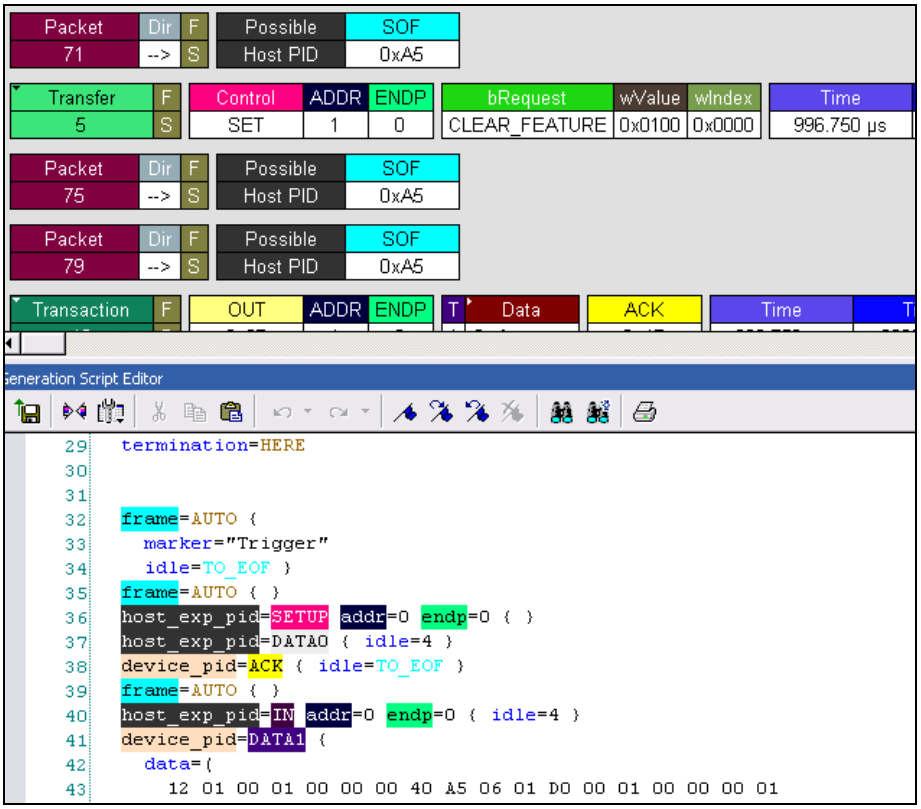


Ilustración 53. Ejemplo de archivo .utg

Antes de comenzar a detallar cada una de las situaciones, debemos comentar que el conexionado de los cables al analizador en el proceso de generación de información ha sido diferente respecto al del proceso de captura. En este segundo caso, requeriremos capturar información al mismo tiempo que la enviamos (ya que el analizador aquí debe actuar como dispositivo y como analizador), por lo que tiraremos un cable hacia el host (como hacíamos antes) y otro desde la parte de captura del analizador (captura) hasta la parte de generación (envío), según la **Ilustración 54**.

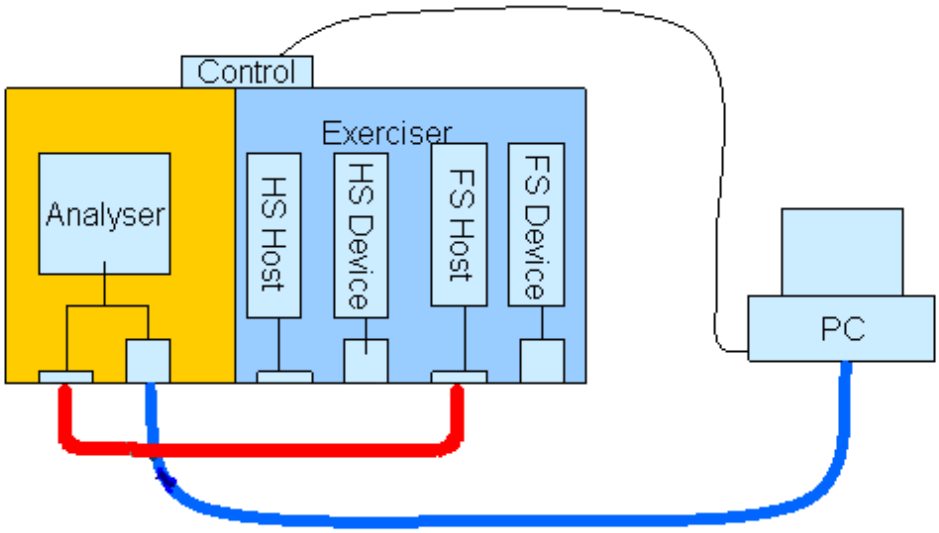


Ilustración 54. Conexión para el generador de tramas

El flujo de información comienza en la parte del generador (*exerciser*) (USB-A), va hacia la parte de captura (*analyser*) (USB-B), ahí concurre con la que viene del host (USB-A), produciéndose aquí la captura de la información.

Los ataques que realizaremos a continuación son los llamados de sustitución o de replay. El hecho de poder suplantar una identidad (y un dispositivo) es un error grave de seguridad. Realizaremos estas pruebas en los dispositivos biométricos que están a prueba y concluiremos con ello nuestros ataques MITM.

8.2.1. Panasonic Authenticam PrivateID

En primer lugar, vamos a comenzar con el ataque a la cámara de Panasonic, para ello explicaremos los pasos que vamos a seguir.

La información capturada por el analizador sobre la cámara contemplaba, como en el resto de dispositivos, los pasos de conexión del dispositivo al host y de autenticación. Toda esta información fue trasladada a un archivo de tipo “.utg” en el que se encuentran instrucciones para responder en el proceso de control (tanto para el setup del dispositivo como para los keep-alive) y la información de la imagen transmitida con anterioridad (imagen que utilizaremos para hacer la suplantación).

Después, pondremos el analizador en el modo de “envío de tramas generadas desde el analizador al host” (**Ilustración 55**) y pondremos el analizador a capturar la información. Tras comprobar que todas las conexiones están correctas comenzamos la comunicación.

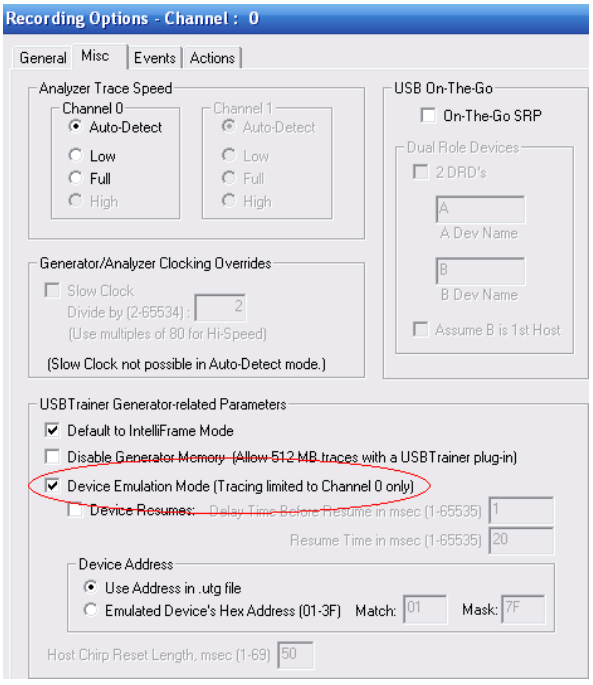


Ilustración 55. Modo emulación de dispositivo

Este proceso emula al de una persona identificándose frente a un sistema de reconocimiento de iris. En teoría, la persona debería estar físicamente colocando su ojo ante la cámara para proceder a su reconocimiento, pero nosotros hemos reducido el escenario a un analizador de protocolo USB y a un ordenador para monitorizar el resultado y controlar el software de la cámara.

Cuando comenzamos a generar tráfico, tenemos el software del laboratorio dispuesto para realizar la autenticación del usuario. En este caso el software no necesita detectar la cámara para inicializarse, por lo que lo podemos ejecutar previamente al envío de tramas desde el analizador. Posteriormente se realizan todos los pasos de detección y configuración del dispositivo con éxito (se observa que el sistema operativo reconoce la cámara) y se envía la información del iris (que antes habíamos capturado) del analizador al host.

La autenticación se realiza con éxito, e incluso se puede llegar a ver en el software del laboratorio la imagen del ojo tal y como lo colocamos cuando realizamos la captura (**Ilustración 56**). Esto quiere decir que hemos completado el ataque MITM con éxito, desde el proceso de captura hasta la manipulación de la información con la posterior suplantación de identidad.

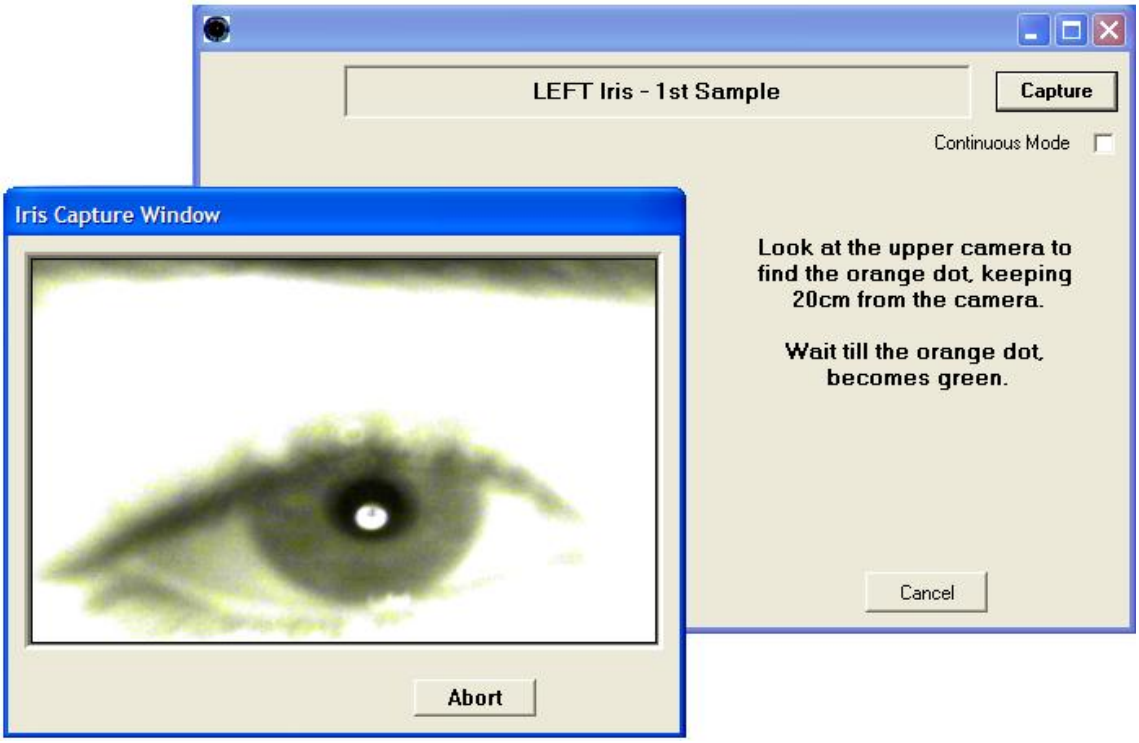


Ilustración 56. Imagen obtenida en el proceso de suplantación

8.2.2. DELL Biometric Coprocessor

Procedemos ahora a realizar el ataque MITM de suplantación sobre el dispositivo de DELL. En la fase de captura no llegamos a conseguir ver la imagen de la huella debido a que se encontraba cifrada, pero de todas formas, disponemos de la información que genera el dispositivo para su configuración y la imagen realmente se envía cifrada. Luego en principio, no hay ningún inconveniente para que completemos el ataque. En este caso, los pasos hasta el comienzo de la transmisión de la información son los siguientes:

En primer lugar, realizamos todo el conexionado y generamos el archivo “.utg” a partir del fichero capturado en la fase anterior. Intentamos ejecutar el software, pero en este caso el sistema operativo necesita que el dispositivo esté conectado para arrancarlo. Por lo tanto, pondremos el analizador a capturar y comenzaremos la comunicación lanzando el archivo generado.

El sistema operativo no reconoce el dispositivo. Observando la captura de la comunicación, vemos que hay tramas que no se han recibido correctamente (el host puede requerir un tipo de trama diferente, dependiendo de la comunicación). Con el fin de solucionar el problema, modificamos el archivo .utg para que responda a las exigencias del Host.

Transfer	F	Control	ADDR	ENDP	Cplt	bRequest	wValue	wIndex
44	S	SET	1	0	NO	CLEAR_FEATURE	DEVICE_REMOTE_WAKEUP	For Device
Transfer	F	Control	ADDR	ENDP	Cplt	bRequest	wValue	wIndex
45	S	SET	1	0	NO	CLEAR_FEATURE	DEVICE_REMOTE_WAKEUP	For Device

Ilustración 57. Error, ataque sobre el dispositivo DELL sin completar

Después de volver a iniciar la comunicación, nos encontramos con que el PC tampoco reconoce el dispositivo, pero tarda un poco más en lanzar el mensaje. En las tramas analizadas podemos observar cómo se ha completado correctamente el proceso de setup, pero poco después de comenzar el proceso de datos, la comunicación se ha suspendido y reseteado.

Esto ha provocado que se reinicie el proceso de setup y se modifiquen las direcciones asignadas al dispositivo, lo cual da lugar a que el dispositivo no encuentre al host y viceversa, y que ya no se completen más transacciones. El proceso no se ha completado debido a que las claves utilizadas no coincidían con las que el host pedía (el analizador ha utilizado las claves caducadas de la sesión anterior ya que no es capaz de generar otras nuevas).

El envío nunca se podrá llevar a cabo en este caso, ya que ni siquiera el sistema operativo reconoce al analizador como el dispositivo DELL. Si nos detenemos en la comunicación capturada podemos comprobar que se han producido varias situaciones de error provocadas por timers agotados (**Ilustración 58**).

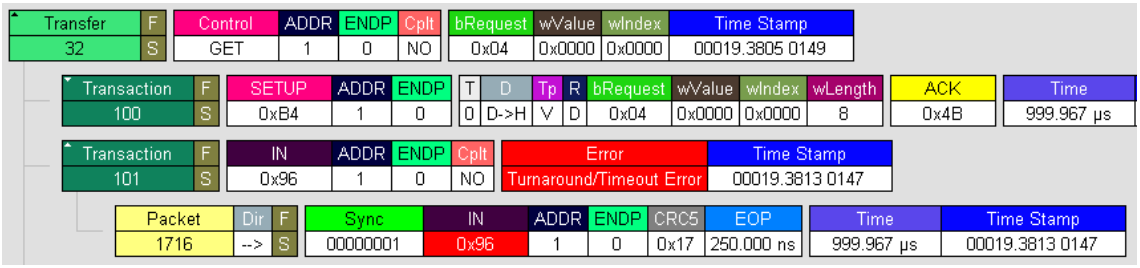


Ilustración 58. Error, timers agotados durante el ataque

Analizando detenidamente el proceso, vemos que el setup se produce correctamente y que se identifica al dispositivo, pero luego la comunicación se suspende (**Ilustración 59**) y parece que hay un problema con las direcciones (ya que hay errores por timers). En este punto llegamos a una conclusión y es que el dispositivo genera claves de sesión con el fin, precisamente, de evitar este tipo de ataques. No conseguiremos suplantar a este dispositivo dado que nuestra clave (que se asigna en el proceso de configuración) nunca será la misma.

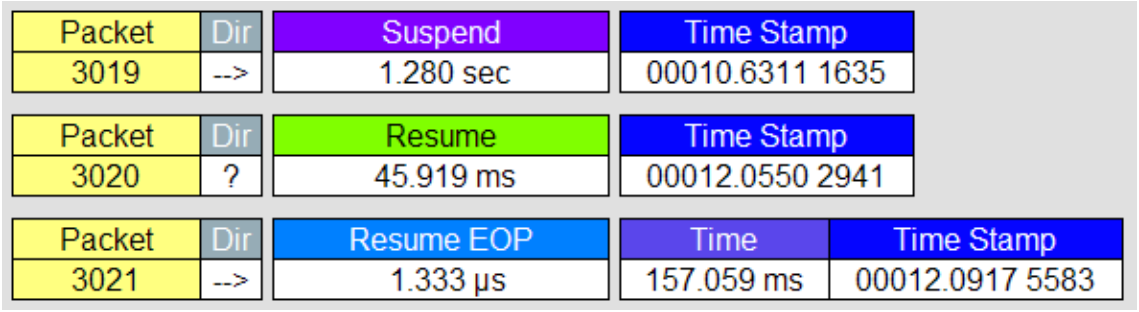


Ilustración 59. Suspensión y resumen de la comunicación

La **clave de sesión** es una clave criptográfica que se usa sólo durante un tiempo limitado. El principio de las claves de sesión es simple: consiste en generar de forma aleatoria una clave de sesión de un tamaño razonable y en cifrar esta clave utilizando un algoritmo de cifrado de clave pública (más precisamente, utilizando la clave pública del receptor). En nuestro caso, el dispositivo DELL utiliza RSA como algoritmo de cifrado. El receptor puede descifrar la clave de sesión con su clave privada. El emisor y el receptor comparten una clave que sólo ellos conocen. Por lo tanto, pueden enviar otros documentos cifrados utilizando un algoritmo de cifrado simétrico.

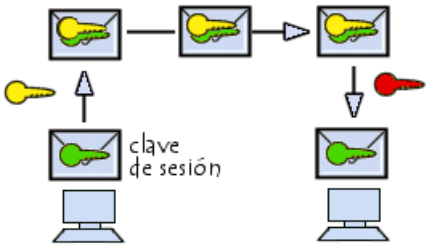


Ilustración 60. Esquema del uso de la clave de sesión [D-H99]

8.2.3. Biometrika FX-3000

Procedemos ahora a realizar el ataque MITM sobre el dispositivo de huella dactilar de Biometrika. Como hemos visto antes, disponemos de una imagen de una huella dactilar tomada durante el proceso de captura estando el nivel de seguridad a cero. Por tanto, llevaremos a cabo dos tipos de ataques. El primero de ellos será con la imagen en claro (sin seguridad) y el segundo con la imagen cifrada. En principio, ambos ataques nos deberían llevar a los mismos resultados ya que en ambos casos, el receptor espera la imagen tal y como la vamos a enviar. Realizaremos este primer proceso con la imagen en claro.

Comenzamos realizando las conexiones como de costumbre, creando el fichero “.utg” con los datos del proceso de captura y colocamos el software de Biometrika en modo seguridad nula. En este caso surge un inconveniente, y es que el archivo “.utg” que se ha generado automáticamente contiene algunos errores que podemos ver en la **Ilustración 61**.

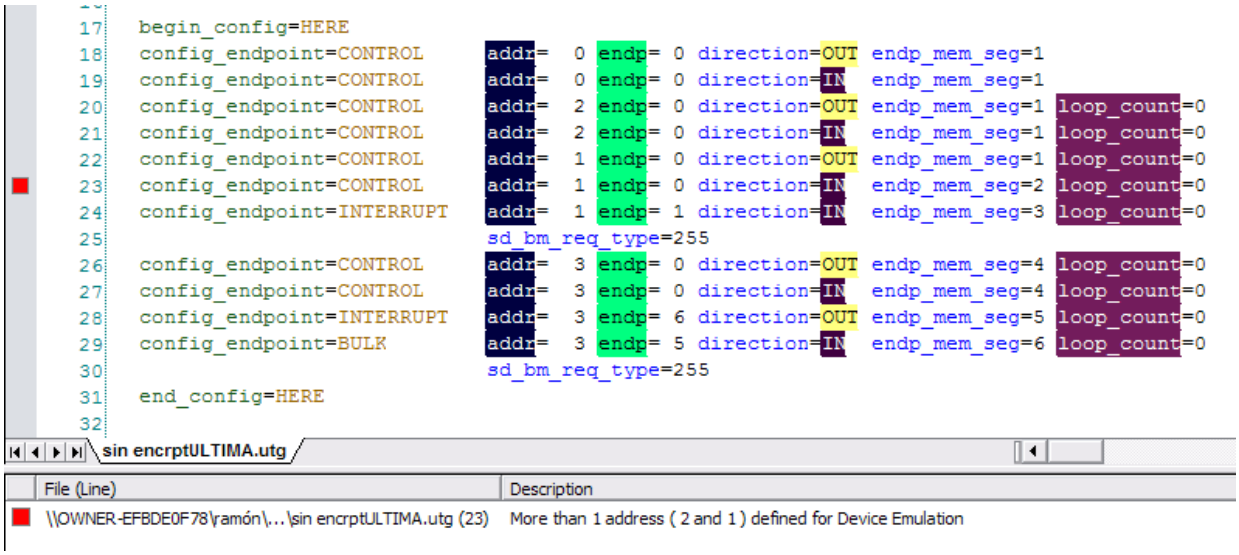


Ilustración 61. Error en el fichero generado por el analizador

El software del analizador ha configurado los endpoints de forma que se le asigna más de una dirección al dispositivo. Como es la primera vez que ocurre esto y no sucede en ningún dispositivo más, no se trata de un error por defecto, sino que se puede tratar de un mecanismo de defensa, relacionado con las claves de sesión, del propio dispositivo. Una vez solucionados los errores del script definiendo una única dirección para el dispositivo, ya podemos obtener el archivo “.utg”.

De nuevo, el software no se ejecuta si no se detecta el dispositivo, por lo que hacemos que el analizador se ponga en modo captura y generación de tráfico. Cuando intentamos arrancar el software, este lanza un error como el del la **Ilustración 62**.

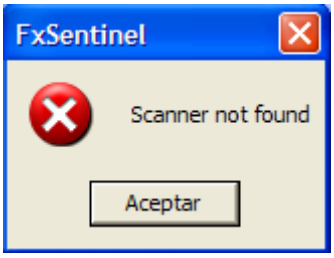


Ilustración 62. Error del escáner de Biometrika

Al igual que en el caso del dispositivo DELL, este hecho se debe a la autogeneración de claves de sesión en cada transferencia, de forma que no sea posible la suplantación. Como hemos visto, el sistema de seguridad del dispositivo de Biométrica utiliza mecanismos de cifrado de 128 bits.

Comprobamos que, tras varias transferencias incompletas, la comunicación se resetea con objeto de completarse. Debido a este sistema de claves nunca se podrá llevar a cabo el ataque MITM.

Transfer	F	Control	ADDR	ENDP	Cplt	bRequest	wValue	wIndex	Time	Time Stamp
29	S	SET	1	0	NO	0x01	0x0001	0x0001	999.967 µs	00026.3414 3359
Transfer	F	Control	ADDR	ENDP	Cplt	bRequest	wValue	wIndex	Time	Time Stamp
30	S	SET	1	0	NO	0x01	0x0001	0x0001	28.328 ms	00026.3422 3357

Packet	Dir	Reset	Time	Time Stamp
5838	-->	31.187 ms	94.668 ms	00026.3649 0511

Ilustración 63. Transferencia incompleta en el ataque al dispositivo de Biometrika

Es curioso comprobar en este caso como, pese a enviar las imágenes sin cifrar, el dispositivo sigue utilizando claves de sesión, lo que resulta un tanto incoherente ya que realmente la imagen se va a enviar en claro. Se está utilizando un sistema de seguridad para después realizar una comunicación sin proteger los datos. Repetimos el proceso con la imagen cifrada pero los resultados son, como ya se esperaba, los mismos.

8.2.4. Fujitsu PalmSecure

El dispositivo de Fujitsu es un caso especial en nuestra investigación. Como ya hemos visto antes, hay un problema de “chirp handshake”, por el cual el dispositivo funciona a velocidades diferentes a las que lo hace el analizador. Esto va a dar lugar a que tengamos problemas para generar tráfico desde el analizador y no podamos completar los ataques MITM. Siguiendo los mismos pasos que en los dispositivos anteriores, llegamos a crear un archivo de tipo “.utg”.

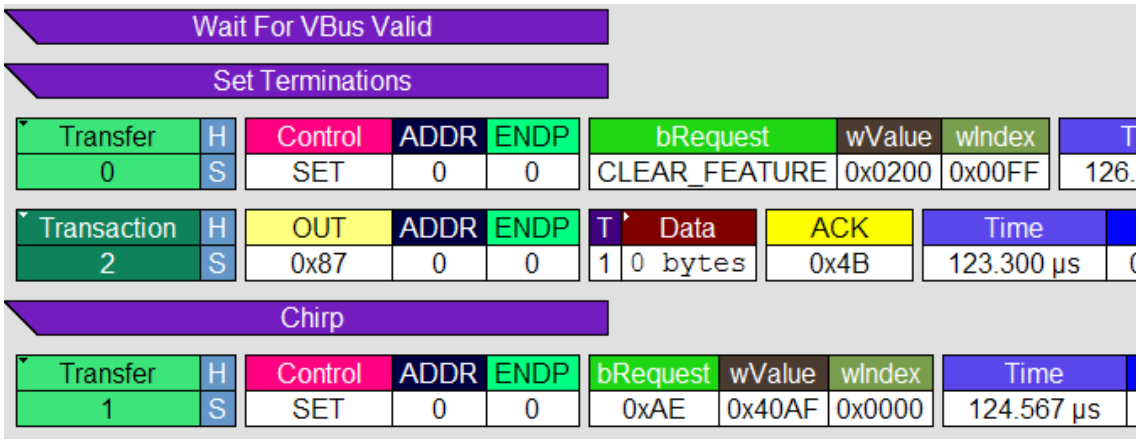
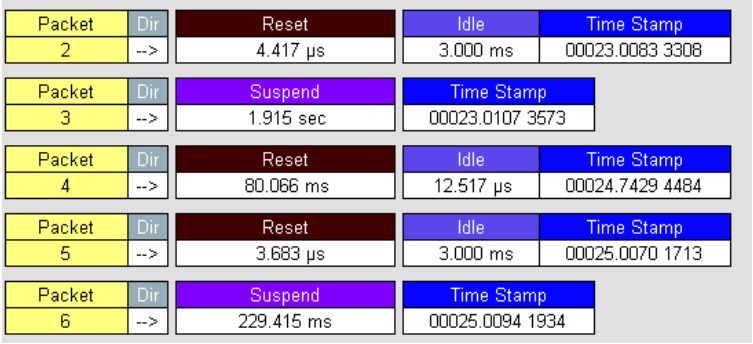


Ilustración 64. Comienzo de archivo .utg para ataque sobre dispositivo de venas

Como podemos observar en la **Ilustración 64**, la comunicación se realiza en modo HS (USB 2.0). Según las especificaciones de nuestro analizador, podemos realizar captura de datos a velocidad HS (por esto mismo hemos podido completar la captura de datos), pero en cuanto a la comunicación en el proceso de generación de tramas, no supera la velocidad FS (USB 1.x). Por lo que podemos generar ficheros que den lugar a una comunicación en HS (como el de la imagen superior), pero en cuanto comenzamos a generar tráfico, el analizador no soporta la comunicación.

El reset y el suspend de forma cíclica (**Ilustración 65**) nos dan a entender la imposibilidad en el envío debido a la incompatibilidad de las velocidades. Futuras actualizaciones del analizador nos permitirán completar el ataque MITM sobre el dispositivo de venas de Fujitsu.



Packet	Dir	Reset	Idle	Time Stamp
2	-->	4.417 µs	3.000 ms	00023.0083 3308
Packet	Dir	Suspend	Time Stamp	
3	-->	1.915 sec	00023.0107 3573	
Packet	Dir	Reset	Idle	Time Stamp
4	-->	80.066 ms	12.517 µs	00024.7429 4484
Packet	Dir	Reset	Idle	Time Stamp
5	-->	3.683 µs	3.000 ms	00025.0070 1713
Packet	Dir	Suspend	Time Stamp	
6	-->	229.415 ms	00025.0094 1934	

Ilustración 65. Suspensión continua de la comunicación con el dispositivo de venas

Capítulo 9:

Fallos de seguridad y líneas futuras

En este capítulo hablaremos de los fallos de seguridad que hemos encontrado en cada dispositivo y comentaremos futuras mejoras posibles que hemos manejado.

9.1 Fallos de seguridad en los dispositivos

En este apartado abordaremos los fallos de seguridad derivados de los ataques de tipo MITM que hemos realizado en el proyecto. Sacaremos una conclusión final del grado de seguridad que ofrece cada dispositivo, en función de su resistencia a las pruebas llevadas a cabo.

9.1.1. Panasonic Authenticam PrivateID

La cámara de iris es el dispositivo que más fallos de seguridad presenta de entre todos los elegidos. Está desprovista de cualquier sistema de seguridad que la proteja de los ataques, por lo que conseguir realizar con éxito el ataque MITM ha sido relativamente sencillo.

Simplemente, siguiendo el mismo comportamiento que vimos en el proceso de captura podemos autenticarnos sin ningún problema en el software de la cámara. Después de generar el archivo .utg, de comenzar a enviar tramas haciéndonos pasar por el dispositivo y de inicializar el software, la autenticación se realiza instantáneamente y con éxito, resultando esto un grave problema de seguridad. Por lo tanto, haciendo un resumen:

9.1.1.1. Ataque de captura de la información

- El dispositivo permite realizar el proceso de captura sin presentar ninguna resistencia. Las imágenes se envían en claro y con el software de tratamiento de imagen podemos obtener perfectamente la foto del iris.

- La monitorización de la información nos permite comprobar cómo se realiza todo el proceso y almacenarlo, de forma que conocemos perfectamente cómo funciona y después podemos utilizarlo en el segundo ataque.

9.1.1.2. Ataque de suplantación de identidad

- El sistema operativo permite que el analizador complete el proceso de setup, reconociéndolo como si fuese realmente la cámara de iris.

- Se realiza el intercambio de datos de forma correcta, reconociendo el software la recepción de la información por parte del analizador como si fuese realmente la cámara.

- La autenticación se realiza de forma instantánea y correcta, del mismo modo que si hubiésemos puesto nuestro propio ojo, completándose así el ataque MITM de forma exitosa.

9.1.2. DELL Biometric Coprocessor

El dispositivo de DELL presenta un sistema de seguridad lo suficientemente fuerte para que nos sea imposible llevar a cabo con éxito los ataques MITM. No obstante, hemos capturado información valiosa del dispositivo (por la cual entendemos cómo funciona su comunicación) y hemos conseguido hacer creer al software por momentos que estaba intercambiando información con el dispositivo y no con el analizador (como era en realidad).

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	Time	Time Stamp
12	S	SET	1	0	SET_CONFIGURATION	New configuration 1	796.977 ms	00008.3495 5694

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Data	Time	Time Stamp
13	S	GET	1	0	0x04	0x0000	0x0000	8 bytes	99.997 ms	00009.1871 4338
14	S	GET	1	0	0x04	0x0000	0x0000	8 bytes	4.000 ms	00009.2671 4167
15	S	SET	1	0	0x0C	0x0100	0x0400	1 byte	4.000 ms	00009.2703 4161

Transfer	F	Bulk	ADDR	ENDP	Bytes Transferred	Time	Time Stamp
16	S	IN	1	1	568	8.000 ms	00009.2735 4154

Transfer	F	Interrupt	ADDR	ENDP	Bytes Transferred	Time	Time Stamp
17	S	OUT	1	2	17	202.994 ms	00009.2799 4140

Ilustración 66. Información de la comunicación obtenida del dispositivo de DELL

Procedemos a hacer un cómputo general de los resultados obtenidos:

9.1.2.1. Ataque de captura de la información

- El dispositivo permite realizar el proceso de captura, pero envía las imágenes obtenidas cifradas, por lo cual no podemos extraer ninguna huella dactilar.

- La monitorización de la información nos permite comprobar cómo se realiza todo el proceso y almacenarlo, de forma que conocemos perfectamente cómo funciona y después podemos utilizarlo en el segundo ataque (aunque no hayamos podido visualizar ninguna imagen).

9.1.2.2. Ataque de suplantación de identidad

- El sistema operativo no permite que el analizador complete el proceso de setup, por lo que ni siquiera se reconoce el dispositivo.

- Se realiza el intercambio de datos de forma correcta hasta completar el proceso de configuración, pero casi de inmediato aparecen errores por timers debido a transferencias incompletas, luego no se permite realizar el ataque.

- Por otra parte, sabemos exactamente en qué punto de la comunicación se produce la situación de suspensión y reset, por lo que en posteriores ataques más perfeccionados ya sabremos por dónde debemos empezar a atacar.

9.1.3. Biometrika FX-3000

Con el dispositivo de Biométrica hemos seguido los mismos pasos que con el de DELL a la hora de realizar los ataques, resultando todos escasamente exitosos. En este caso tenemos la oportunidad de enviar la imagen en claro (que obtuvimos en la captura), pero el software de Biométrica necesita del dispositivo para ser lanzado y por tanto, de esta forma nunca podremos ejecutarlo.

Podemos sacar unas conclusiones semejantes:

9.1.3.1. Ataque de captura de la información

- Cuando seleccionamos un nivel de seguridad mayor que cero, el dispositivo permite realizar el proceso de captura, pero envía las imágenes obtenidas cifradas, por lo cual no podemos extraer ninguna huella dactilar.

- Con nivel de seguridad nulo, podemos obtener la imagen de la huella dactilar de forma nítida, con lo que conocemos cómo son las imágenes que toma el dispositivo.

- La monitorización de la información nos permite comprobar cómo se realiza todo el proceso y almacenarlo, de forma que conocemos perfectamente cómo funciona y después podemos utilizarlo en el segundo ataque.

9.1.3.2. Ataque de suplantación de identidad

- El PC permite que el analizador complete el proceso de setup. Aunque al final no se reconozca el dispositivo, el PC sí que ha interpretado por un instante que se estaba comunicando con el dispositivo de huella dactilar. No así el software, que nunca se ejecuta debido a que no se detecta actividad del dispositivo.

- Conocemos en qué punto de la comunicación se produce la situación de suspensión y reset (**Ilustración 67**), por si en un futuro encontramos una solución al problema.

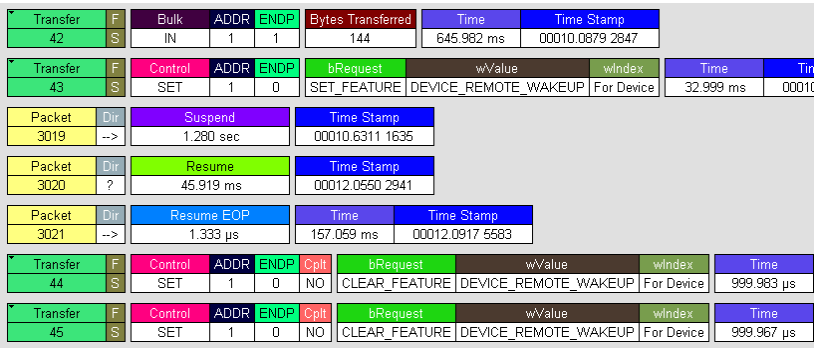


Ilustración 67. Suspensión de la comunicación con el dispositivo de Biometrika

9.1.4. Fujitsu PalmSecure

El dispositivo de reconocimiento de venas de la mano nos ha permitido completar el proceso de captura, aunque la imagen obtenida no revela la zona que buscábamos (el mapa de venas). Pero no nos ha permitido ni siquiera intentar realizar el ataque de suplantación debido al problema de incompatibilidad de velocidades comentado anteriormente.

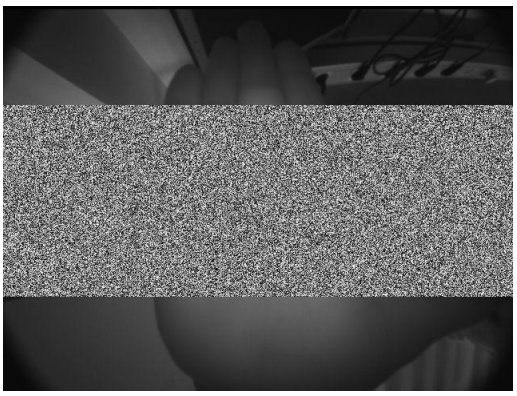


Ilustración 68. Imagen obtenida en el análisis del dispositivo de venas

Resumiendo, las conclusiones sacadas han sido:

9.1.4.1. Ataque de captura de la información

- El dispositivo permite realizar el proceso de captura sin presentar ninguna resistencia. Las imágenes aunque no estén completas, pueden servirnos de ayuda para hallar el mapa de venas en un futuro.

- La monitorización de la información nos permite comprobar cómo se realiza todo el proceso y almacenarlo, de forma que conocemos perfectamente cómo funciona y después podemos utilizarlo en el segundo ataque.

9.1.4.2. Ataque de suplantación de identidad

En este caso, como ya hemos visto, cualquier intento de ataque por suplantación con el equipamiento disponible es imposible de llevar a cabo por la incompatibilidad de velocidades entre emisor y receptor.

9.2. Líneas futuras de trabajo

Las posibles mejoras en este proyecto siguen tres corrientes bien diferenciadas. Por un lado está la mejora de cara al dispositivo de venas de la mano. Por otro lado está la posibilidad de realizar un mayor control sobre los scripts para realizar los ataques con el fin comprobar si en un futuro podríamos generar tramas dinámicamente. Por último, esta la posibilidad de utilizar otro analizador de protocolo pero sobre comunicaciones Ethernet.

Como ya hemos comentado en varias ocasiones anteriormente, el dispositivo de venas de la mano de Fujitsu tiene un problema de incompatibilidad de velocidades con el analizador. Esto se debe básicamente, a que funciona en modo Hi-Speed. El analizador (con la licencia actual) soporta el modo Hi-Speed en su modo trainer (captura), pero no en el modo exerciser (generación de tramas), por lo que no hemos podido completar el ataque MITM sobre este dispositivo. En cuanto se obtenga la licencia necesaria para modificar la frecuencia del analizador en el modo de envío de tramas, podremos completar los ataques sobre el dispositivo.

Como otra mejora, también hemos contemplado la posibilidad de desarrollar scripts que sean capaces, de forma dinámica, de generar una respuesta correcta ante cualquier petición del Host. Estos scripts no serían desarrollados automáticamente por el analizador (sino por nosotros mismos) y podrían generar, por ejemplo, claves de sesión acorde con las que requiriese el Host.

En último lugar, en un principio se pensaba realizar el proyecto sobre un analizador de protocolo Ethernet (**Ilustración 69**), pero siendo más amplia la gama de dispositivos USB que de Ethernet en el laboratorio, nos decantamos finalmente por el analizador de tráfico USB. Habiendo concluido este estudio, no será inconveniente en un futuro realizar estos mismos ataques sobre dispositivos de tipo Ethernet, aprovechándonos de la experiencia adquirida sobre el analizador USB.

Ilustración 69. Analizador Ethernet Compass CN-100 [Com09]

Capítulo 10:

Bibliografía

- **[Akj04]:** Anil K. Jain, "*Handbook of multibiometrics*", Springer, 2004.

- **[Ace09]:** Medidas biométricas de seguridad en la empresa Acens. http:/www.acens.com/infraestructura

- **[Ama07]:** Revista Savia Amadeus, núm.47, **"***Sociedad Aeropuertos*, mayo 2007. Medidas biométricas de seguridad en diferentes aeropuertos europeos. http://www.es.amadeus.com/es/documents/aco/spain/es/revista_savia/aeropuertos47.pdf

- **[Ber09]:** Alphonse Bertillon, *"Alphonse Bertillon's Instructions for Taking Descriptions for the Identification of Criminals and Others, by Means of Anthropometric Indications"*, Kessinger Pub, 2009.

- **[Bio09]:** Página web de Biometrika. http://www.biometrika.it

- **[Com09]:** Página web del analizador Ethernet Compass Networks. http://www.compassnettest.com

- **[D-H99]:** Página web del IETF sobre clave de sesión de Diffie-Hellman. http://www.ietf.org/rfc/rfc2631.txt

- **[Del09]:** Página web de DELL. http://www.dell.com/es

- **[Fuj09]:** Página web de Fujitsu.

  http://www.fujitsu.com

- **[Gal83]:** Francis Galton, *"Inquiries into Human Faculty and Its Development"*, Primera edición, *Macmillan*, 1883.

- **[Ing00]:** Página web Universidad de Chile.

  http://www2.ing.puc.cl/~iing/ed429/sistemas_biometricos.htm

- **[Iwr02]:** Revista Iworld, Pedro Guillén, "*Biometría, cuando el cuerpo es la evidencia*", Diciembre 2002.

- **[Jpg07]:** Página del grupo JPEG.

  http://www.jpeg.org/jpeg/index.html

- **[Lec09]:** Página web del fabricante del analizador USB (Lecroy).

  http://www.lecroy.com

- **[Nec09]:** NEC Electronics. USB 2.0 Basic specification.

  http://www.necel.com/usb

- **[Ow09]:** The Open Web Application Security Project (OWASP).

  http://www.owasp.org

- **[Pan09]:** Página web de Panasonic.

  http://www.panasonic.es

- **[Pc08]:** Asignatura "Percepción Computacional", Ingeniería informática 2008/2009, Universidad Carlos III de Madrid.

- **[Usb09]:** Especificación de USB.

  http://www.usb.org

- **[Usb3.0]:** Avance de la especificación 3.0 de USB.

  http://www.usb.org/developers/ssusb